HTML PRÁCTICA 1

Ejercicio 1

1)Para publicar información y distribuirla globalmente, se necesita un lenguaje entendido universalmente, una especie de lengua franca de publicación que todas las computadoras puedan comprender potencialmente. El lenguaje de publicación usado por la World Wide Web es el HTML (acrónimo de HyperText Markup Language, Lenguaje para el Formato de Documentos de Hipertexto).

El HTML da a los autores las herramientas para:

- Publicar documentos en línea con encabezados, textos, tablas, listas, fotos, etc.
- Obtener información en línea a través de vínculos de hipertexto, haciendo clic con el botón de un ratón.
- Diseñar formularios para realizar transacciones con servicios remotos, para buscar información, hacer reservas, pedir productos, etc.
- Incluir hojas de cálculo, videoclips, sonidos, y otras aplicaciones directamente en sus documentos.

2)

- Interoperabilidad: Los documentos deben ser compatibles con diferentes navegadores, plataformas y dispositivos, asegurando que puedan ser interpretados correctamente por cualquier agente de usuario.

- Accesibilidad: El contenido debe ser accesible para personas con discapacidades, siguiendo pautas como las WCAG (Pautas de Accesibilidad para el Contenido Web).

- Estructuración lógica y presentación separada:

- Usar HTML para definir la estructura semántica del documento (encabezados, párrafos, listas, etc.).

- Emplear CSS para controlar la presentación visual (colores, fuentes, diseños), evitando el uso de etiquetas HTML con fines de formato (como <font> o atributos como align).

- Compatibilidad con versiones anteriores: Asegurar que los documentos funcionen correctamente en navegadores antiguos, aunque aprovechen las características de HTML 4.01.

- Validación: Crear documentos que cumplan con la gramática formal de HTML 4.01 (DTD), lo que facilita la interoperabilidad y el mantenimiento.

3) Un elemento o atributo está desaprobado cuando todavía forma parte del lenguaje, pero ya no se recomienda su uso porque existe una alternativa mejor. Los navegadores siguen

soportándolo para mantener compatibilidad con sitios viejos, pero se espera que los desarrolladores eviten usarlo.
Un elemento o atributo es obsoleto cuando ya no está definido en la especificación actual y no se garantiza que los navegadores lo sigan soportando. Básicamente, quedó fuera del estándar y no debería usarse.

4) El DTD en HTML es una declaración que se incluye al comienzo de un documento HTML para indicar qué versión del lenguaje se está utilizando y cuáles son las reglas que debe seguir el contenido para ser considerado válido. Sirve como una guía para que los navegadores interpreten correctamente las etiquetas, y también permite a los validadores verificar si el documento cumple con los estándares establecidos.

Existen diferentes tipos de DTD:

- HTML 4.01 Strict: Incluye solo los elementos y atributos permitidos por el estándar y no admite el uso de características desaprobadas. Es ideal para quienes buscan escribir un código limpio, moderno y preparado para el futuro.
- HTML 4.01 Transitional (transicional): Permite el uso de elementos y atributos desaprobados. Fue pensado como una forma de facilitar la migración desde versiones anteriores de HTML, ofreciendo mayor flexibilidad aunque con menor rigor en cuanto a las buenas prácticas.
- HTML 4.01 Frameset: Similar al Transitional pero además permite el uso de marcos (frames), una técnica utilizada para dividir la ventana del navegador en varias secciones independientes.

5) Los metadatos proporcionan información estructurada sobre el documento, como:

- Su título.
- Su tipo de contenido (ej: text/html).
- Su codificación de caracteres (ej: UTF-8).
- Relaciones con otros recursos (ej: hojas de estilo o scripts).
- Instrucciones para navegadores y motores de búsqueda.

Los metadatos se especifican en <head> usando <title>, <meta>, <link> y <base>. Su correcta implementación asegura que el documento cumpla con los estándares del W3C y sea procesable por máquinas y humanos.

Ejercicio 2

a. Representa un comentario en HTML. Este tipo de elemento se puede colocar en cualquier sección del documento, tanto dentro de <head> como en <body>, y su función es únicamente informativa, ya que no produce ningún efecto visible en el navegador. No utiliza etiquetas ni atributos, y su uso no es obligatorio, aunque resulta útil para documentar el código.

b. Define un elemento contenedor mediante la etiqueta , el cual se utiliza normalmente para agrupar bloques de contenido dentro del <body> del documento. El atributo presente es id=”bloque1” que identifica de forma única ese contenedor, aunque su uso no es obligatorio. La etiqueta posee una etiqueta de apertura y una de cierre, y puede contener otros elementos HTML.
c. Representa un elemento de imagen que debe colocarse en el <body> del documento. Se trata de un elemento vacío, es decir, que no requiere etiqueta de cierre. El atributo src indica la ruta de la imagen y es obligatorio para que se muestre correctamente, aunque en este caso está vacío. El atributo alt también es obligatorio para accesibilidad, ya que proporciona un texto alternativo. Los demás atributos son opcionales, y aportan identificación, dimensiones o una descripción larga de la imagen.
d. Corresponden a elementos de metadatos, los cuales deben colocarse dentro del <head> del documento. Estos elementos son vacíos y no se visualizan en el navegador. El primer meta incluye atributos como name, lang y content, siendo este último obligatorio. El segundo meta utiliza el atributo http-equiv para simular una cabecera HTTP y también contiene un content obligatorio. Ambos proporcionan información sobre la página al navegador y a motores de búsqueda.
e. Define un hipervínculo, y se coloca dentro del <body>. Utiliza las etiquetas <a> de apertura y cierre, y contiene varios atributos. El más importante es href, que indica la URL de destino y es obligatorio. Los demás son opcionales, y proporcionan información adicional sobre el tipo de contenido, el idioma, la codificación y la relación del recurso enlazado con la página actual. Al hacer clic en el texto “Resumen HTML”, se abre la página indicada.
f. Representa una tabla HTML, que debe ubicarse dentro del <body>. La tabla define un resumen (summary) y un ancho (width), ambos atributos opcionales. El <caption> define un título para la tabla y el atributo align="top" indica su posición. Dentro de las filas (<tr>), se utilizan <th> para encabezados y <td> para celdas de datos, y algunos encabezados incluyen el atributo scope para mejorar la accesibilidad, especificando si se refieren a una columna o una fila. Ninguno de los atributos es obligatorio, pero ayudan a organizar y presentar datos de forma estructurada.

Ejercicio 3

3.a) Enlaces (<a>)
Los segmentos presentan diferentes comportamientos y atributos en los enlaces. El primer enlace dirige a Google en la misma pestaña, mientras que el segundo lo abre en una nueva pestaña gracias al atributo target="_blank". El tercer enlace está incompleto pero incluye metadatos como type, hreflang, charset y rel, que proporcionan información adicional sobre el recurso enlazado. El cuarto enlace utiliza href="#", lo que lo convierte en un enlace nulo, comúnmente usado para activar funciones JavaScript. El quinto enlace es un ancla interno que redirige a un punto específico de la página, definido por el sexto elemento, que actúa como marcador usando los atributos name e id.

3.b) Imágenes dentro de enlaces (<a><img>)
En este caso, las diferencias radican en cómo se estructuran los enlaces y las imágenes. El primer segmento muestra la imagen y el enlace como elementos separados dentro de un párrafo. El segundo segmento envuelve solo la imagen con el enlace, dejando el texto "Click aquí" fuera. El tercer segmento incluye tanto la imagen como el texto dentro del mismo enlace, lo que hace que ambos sean clickeables. El cuarto segmento repite el enlace tanto para la imagen como para el texto, lo que resulta redundante pero funcional. Cada enfoque afecta la interactividad y la accesibilidad del contenido.

3.c) Listas y bloques de texto
Aquí se comparan diferentes métodos para crear listas. El primer segmento usa una lista desordenada (<ul>), el segundo una lista ordenada (<ol>), y el tercero divide la lista ordenada en dos partes, reiniciando la numeración con li value="2". El cuarto segmento simula una lista usando saltos de línea (
) dentro de un bloque de texto (<blockquote>). Cada método tiene implicaciones para la semántica y el estilo, siendo las listas HTML más adecuadas para estructurar contenido enumerado.

3.d) Tablas con estilos diferentes
Las tablas muestran diferencias en cómo se aplican los estilos y la semántica. La primera tabla usa etiquetas <th> para los encabezados, lo que es semánticamente correcto. La segunda tabla utiliza celdas normales (<td>) con estilos en línea y la etiqueta para simular encabezados, lo que es menos ideal para accesibilidad. Ambas tablas logran un diseño similar, pero la primera es más adecuada para datos tabulares estructurados.

3.e) Tablas con celdas combinadas
Estos segmentos ilustran el uso de colspan y rowspan para combinar celdas. La primera tabla usa colspan="3" para un título centrado, mientras que la segunda aplica colspan y rowspan en diferentes celdas para crear un diseño más complejo. Estas técnicas permiten layouts personalizados pero requieren cuidado para mantener la estructura coherente.

3.f) Formularios con diferentes métodos
Los formularios varían en estructura y atributos. El primer formulario usa method="post" y agrupa campos con <fieldset>, ideal para envíos seguros de datos. El segundo formulario omite el método (por defecto es GET) y usa etiquetas <label> para mejorar la accesibilidad. El tercer formulario envía datos por correo (mailto:) pero tiene errores de sintaxis. Cada enfoque tiene ventajas según el contexto de uso.

3.g) Botones con diferentes estilos
El primer botón incluye una imagen y texto enriquecido con HTML, ofreciendo más flexibilidad visual. El segundo botón es un simple botón de texto con el atributo value. El primero es útil para diseños personalizados, mientras que el segundo es más sencillo y directo.

3.h) Radio buttons agrupados
Los primeros radio buttons comparten el mismo nombre (name="opcion"), lo que permite seleccionar solo uno a la vez. Los segundos tienen nombres diferentes (name="opcion1" y name="opcion2"), permitiendo seleccionar ambos, lo que rompe la funcionalidad típica de los radio buttons.

3.i) Listas desplegables
La primera lista desplegable (<select>) permite una sola selección y agrupa opciones con <optgroup>. La segunda lista incluye multiple="multiple", permitiendo selecciones múltiples. La primera es ideal para opciones exclusivas, mientras que la segunda sirve para seleccionar varios items simultáneamente

3.j) Grupos de Radio Buttons
En el primer grupo de radio buttons, ambos elementos comparten el mismo atributo name="opcion", lo que significa que funcionan como un grupo lógico donde solo se puede seleccionar una opción a la vez. Cada radio button tiene su propio id y value, lo que permite identificarlos de manera única tanto para el usuario como para el procesamiento del formulario. Este es el comportamiento estándar y correcto de los radio buttons en HTML.

El segundo grupo, sin embargo, utiliza nombres diferentes para cada radio button (name="opcion1" y name="opcion2"). Esto hace que el navegador los trate como elementos independientes, permitiendo que ambos puedan estar seleccionados al mismo tiempo. Esto va en contra del propósito fundamental de los radio buttons, que es ofrecer una selección exclusiva entre opciones mutuamente excluyentes. Es un error común que puede afectar la usabilidad y la lógica del formulario.

3.k) Listas Desplegables
La primera lista desplegable es un elemento <select> tradicional que permite una sola selección. Utiliza etiquetas <optgroup> para agrupar visualmente las opciones "Mayo" y "Junio" bajo los títulos "Caso 1" y "Caso 2". Este enfoque es ideal cuando se necesita organizar muchas opciones en categorías claras, mejorando la experiencia del usuario.

La segunda lista incluye el atributo multiple="multiple", que transforma el control en una lista de selección múltiple. Además, el nombre name="lista[]" (con corchetes) es particularmente útil cuando se trabaja con lenguajes del lado del servidor como PHP, ya que facilita el manejo de múltiples valores seleccionados como un array. Los mismos grupos de opciones están presentes, pero ahora el usuario puede seleccionar varios elementos simultáneamente manteniendo presionada la tecla Ctrl (o Command en Mac) mientras hace clic..