investigacion

Las etiquetas HTML más comunes incluyen etiquetas para texto y tipografía como <h1> a <h6> para encabezados, <p> para párrafos, <br> para saltos de línea, y <strong> para texto en negrita. También son importantes las etiquetas para enlaces (<a>), imágenes (<img>), listas (<ul>, <ol>, <li>), formularios (<form>) y estructuración de la página (<div>, <body>, <header>, <footer>).

Estructura básica del documento:

* <!DOCTYPE html>: Define el tipo de documento.
* <html>: Etiqueta raíz del documento.
* <head>: Contiene metadatos sobre el documento, como el título.
* <body>: Contiene el contenido visible de la página.

Encabezados:

* <h1> - <h6>: Define encabezados de diferentes niveles.

Párrafos y saltos de línea:

<p>: Define un párrafo, <br>: Inserta un salto de línea, <hr>: Inserta una línea horizontal.

Formatos de texto:

* <strong>: Define texto importante (generalmente en negrita).
* <em>: Define texto enfatizado (generalmente en cursiva).
* <b>: Define texto en negrita (menos semántico que <strong>).
* <i>: Define texto en cursiva (menos semántico que <em>).
* <u>: Subraya texto.
* <small>: Define texto pequeño.
* <big>: Define texto grande.
* <sub>: Define texto como subíndice.
* <sup>: Define texto como superíndice.

Enlaces:

* <a>: Define un hipervínculo. Se usa con el atributo href para especificar la URL de destino.

Imágenes:

* <img>: Inserta una imagen. Se usa con el atributo src para especificar la ruta de la imagen.

Listas:

<ul>: Define una lista desordenada, <ol>: Define una lista ordenada, <li>: Define un elemento de la lista.

Formularios:

* <form>: Define un formulario HTML.
* <input>: Define un campo de entrada (texto, botón, etc.).
* <textarea>: Define un campo de texto multilinea.
* <button>: Define un botón.
* <label>: Define una etiqueta para un elemento de formulario.
* <select>: Define un menú desplegable.
* <option>: Define una opción en un menú desplegable.

Otros elementos de estructura:

* <div>: Define una sección o división en el documento.
* <span>: Define una pequeña sección en línea dentro de un elemento.
* <header>: Define la cabecera de una sección o página.
* <footer>: Define el pie de página de una sección o página.
* <nav>: Define una sección de navegación.
* <article>: Define un artículo independiente.
* <aside>: Define contenido que es tangencial al contenido principal.
* <section>: Define una sección en un documento.

Etiquetas semánticas (HTML5):

* <header>: Define la cabecera de una sección o página.
* <nav>: Define una sección de navegación.
* <main>: Define el contenido principal de un documento.
* <article>: Define un artículo independiente.
* <aside>: Define contenido que es tangencial al contenido principal.
* <footer>: Define el pie de página de una sección o página.
* <figure>: Define una unidad de contenido independiente, como una imagen, un diagrama, etc.
* <figcaption>: Define un título para un elemento <figure>.
* <time>: Define una fecha u hora.

1. color: Define el color del texto.
2. font-family: Define la familia tipográfica del texto.
3. font-size: Define el tamaño del texto.
4. font-style: Define el estilo de la fuente (normal, cursiva, etc.).
5. font-weight: Define el grosor de la fuente (normal, negrita, etc.).
6. text-align: Define la alineación horizontal del texto.
7. text-decoration: Añade subrayado, tachado, etc. al texto.
8. background-color: Define el color de fondo del elemento.
9. background-image: Define una imagen de fondo para el elemento.
10. width: Define el ancho del elemento.
11. height: Define la altura del elemento.
12. padding: Define el espacio entre el contenido y el borde del elemento.
13. margin: Define el espacio alrededor del elemento.
14. border: Define el borde del elemento (color, estilo, ancho).
15. border-radius: Define el redondeo de las esquinas del borde.
16. display: Define cómo se muestra el elemento (bloque, en línea, etc.).
17. position: Define la posición del elemento (relativa, absoluta, fija, etc.).
18. top, right, bottom, left: Define la posición del elemento en relación a su contenedor, cuando se usa position.
19. opacity: Define la transparencia del elemento.
20. visibility: Define si el elemento es visible o no.

El Document Object Model (DOM) es una representación en memoria de la estructura de un documento HTML o XML, que permite a los lenguajes de programación, como JavaScript, acceder y manipular dinámicamente el contenido, la estructura y el estilo de la página web. En esencia, el DOM actúa como un puente entre el código y la página web, permitiendo la creación de páginas interactivas y aplicaciones web complejas

¿Para qué se usa el DOM?

El DOM se utiliza principalmente para:

* **Acceder a elementos de la página:**

Permite identificar y seleccionar elementos HTML específicos (como párrafos, imágenes, divs, etc.) mediante sus etiquetas, IDs, clases, etc.

* **Modificar contenido:**

Una vez que se ha accedido a un elemento, se puede cambiar su contenido textual, atributos (como la URL de una imagen) o añadir o eliminar elementos hijos.

* **Modificar estilo:**

Permite cambiar las propiedades de estilo de los elementos, como el color, el tamaño de la fuente, la posición, etc.

* **Gestionar eventos:**

El DOM permite adjuntar controladores de eventos (como clics, movimientos del ratón, pulsaciones de teclas) a los elementos, de modo que se puedan ejecutar acciones específicas cuando ocurran esos eventos.

* **Crear aplicaciones web dinámicas:**

Al permitir la modificación del contenido y la estructura de la página en tiempo real, el DOM es fundamental para la creación de aplicaciones web interactivas, como redes sociales, juegos online, o cualquier página que requiera actualizaciones dinámicas.

En resumen, el DOM es una herramienta esencial para el desarrollo web moderno, ya que proporciona la interfaz necesaria para interactuar con el contenido de una página web de forma dinámica y crear experiencias interactivas.