Programacion ( Lista para el Debate )

Las 50 etiquetas mas usadas de HTML:

1. <DOCTYP: Define el tipo de documento: <¡DOCTYPE HTML>
2. <html>: Raíz del documento HTML: <html lang=”Es”>...</html>
3. <head>: Contiene metadatos: <head><title>Misitio</title></head>
4. <title>: Titulo de la pagina: <title>Portafolio</title>
5. <meta>: Metadatos como charset o descripción:<meta charset=”UTF-8”>
6. <link>: Enlaza recursos externos: <link rel=”stylesheet” href=”style.css”>
7. <style>: CSS interno: <style>body{ font-family:sans-serif;}<
8. <script>: Codigo JavaScript: <script src=”main.js”></script>
9. <body>: Contenido visible: <body>...</body>
10. <header> Encabezado de sección o página: <header><h1>Blog</h1></header>
11. <nav>: Menu de navegacion: <nav><ul><li>inicio</li></ul></nav>
12. Main: Contenido principal: <main><article>..</article></main>
13. <Section>: Agrupa contenido tematico: <section><h2>Servicios</h2></section>
14. <article> Contenido independiente <article><h2>Post</h2></article>
15. <aside>: Contenido complementario: <aside>Publicidad</aside>
16. <footer>: Pie de página© 2025
17. <h1> a <h6>: Encabezados: <h2>Sobre mi</h2>
18. <p>: Párrafo: <p>Hola mundo</p>
19. <br>: Salto de línea: Hola<br>Mundo
20. <hr> Linea horizontal: <hr>
21. <strong>: Texto importante: <strong>Urgente</strong>
22. <em>: Énfasis: <em>importante</em>
23. <mark>: Texto resaltado <mark>Nuevo</mark>
24. <small>: Texto pequeño <small>Nota legal</small>
25. <blockquote> Cita larga <blockquote>Texto citado</blockquote>
26. <span>: Contenedor en línea <span class=”destacado”>texto</span>
27. <a> Enlace: <a href=https://ejemplo.com”>Visitame</a>
28. <img> imagen: <img src=”foto.jpg” alt=”Foto”>
29. <figure> Agrupa imagen y descripción <figure><img src=”img.jpg”><figcaption>descripcion</figcaption></figure>
30. <figcaption> Pie de imagen: <figcaption>Texto</figcaption>
31. <div>:Define una division o sección generica en un documento: <div><h2>Mi seccion</h2><p>Contenido de la sección.</p></div>
32. <ul> Crea una lista dosordenada de elementos: <ul><li>Elemento 1</li><li>Elemento 2</li></ul>
33. <ol> Crea una lista ordenada de elementos: <ol><li>Primer paso</li><li>Segundo paso</li></ol>
34. <li> Representa un elemento dentro de una lista: <ul><li>Item de lista</li></ul>
35. <dl> Define una lista de descripcion:<dl><dt>Termino</dt><dd>Descripción del termino</dd></dl>
36. <dt> Define un termino o nombre de en una lista de descripcion: <dl><dt>CSS</dt><dd>Hoja de estilos en cascada...</dd></dl>
37. <dd> Define la descripcion o definicion de un termino en un alista de descripcion: <dl><dt>CSS</dl><dd>Hoja de estilos en cascada</dd></dl>
38. <table> Crea una tabla para mostrar datos tabulados: <table><tr><td>Dato 1</td></tr></table>
39. <thread> Agrupa el contenido del encabezado en una tabla: <table><thread><tr><th>Encabezado</th></tr></thread></table>
40. <tbody> Agrupa el contenido del cuerpo en una tabla: <table><tbody><tr><td>Cuerpo</td></tr></tbody></table>
41. <tfoot>: Agrupa el contenido del pie en una tabla: <table><tbody><tr><td>Cuerpo</td></tr></tbody></table>
42. <tr> Define una fila en una tabla: <table><tr><td>celda</td></tr></table>
43. <th> Define una celda de encabezado en una tabla: <table><tr><th>Producto</th><th>Precio>/th></tr></table>
44. <td> Define una celda de datos estandar en una tabla: <table><tr><td>Manzanas</td><td><td>1.50$</td></tr></table>
45. <caption> Define el titulo o leyenda de una tabla: <table><caption>Ventas Mensuales</caption><tr><td>...</td></tr>
46. <form> Define un formulario HTML para la entrada del usuario: <form action=”/submit” method= “post”><input type=”text”></form>
47. <input>: Crea un campo de entrada interactivo para formularios: <input type= “text” name=”nombre” placeholder=”Tu nombre”>
48. <textarea> Define un area de texto multilinea para la entrada del usuario: <textarea name=”comentarios” rows=”4” cols=”50”></textarea>
49. <button> Define un boton cliqueable: <button type=”submit”>Enviar</button>
50. <select> Define una opcion dentro de un elemento: <select><option> value=”us”>Estados unidos</option></select>

Parte 2: CSS 20 formas de añadir estilo a HTML

1. Color

Que hace: Establece el color del texto de un elemento  
Ejemplo de uso: p { color: #336699; }  
Se suele aplicar a: Practicamente cualquier elemento que contenga texto, como parrafos (<p>), encabezados (<h1> a <h6> ), enlaces (<a>) etc

1. Font-family

Que hace: Define el tipo de fuente para el texto de un elemento. Puedes especificar varias fuentes como respaldo

Ejemplos de uso: body { font-family: “Arial”, sans-serif; }

Se suele aplicar a: body (para establecer una fuente global),párrafos, encabezados y cualquier elemento de texto.

1. Font-size

Que hace: Ajusta el tamaño del texto. Se pueden usar unidades como px, em, rem, %

Ejemplo de uso: h1 { font-size: 2.5em; }

Se suele aplicar a: Encabezados, párrafos, elementos de lista (<li>), y cualquier elemento con contenido de texto.

1. Font-weight

Que hace: Define el grosor o la intensidad de la fuente (por ejemplo, normal, negrita).

Ejemplo de uso: strong { font-weight: bold; } o p { font-weight: 500; }

Se suele aplicar a: Elementos de texto, especialmente para enfatizar ciertas palabras o frases.

1. Text-align

Que hace: Alinea el texto dentro de un elemento (izquierda, derecha, centro, justificado).

Ejemplo de uso: h2 { text-align: center; }

Se suele aplicar a: Bloques de texto como párrafos, encabezados, y contenedores como div

1. Line-height

Que hace: Establece la altura de una linea de texto. Afecta el espacio vertical entre lineas.

Ejemplo de uso: p { line-height: 1.6; }

Se suele aplicar a: Parrafos y cualquier bloque de texto para mejorar la legibilidad

1. Background-color

Que hace: Establece el color de fondo de un elemento.

Ejemplo de uso: div { background-color: lightblue; }

Se suele aplicar a: Practicamente cualquier elemento, incluyendo body, div, section, button, etc

1. Background-image

Que hace: Define una imagen como fondo de un elemento.

Ejemplo de uso: body { background-image: url (‘pattern.png’); }

Se suele aplicar a: imagenes, div, botones, inputs,tablas y otros elementos que se benefician de un contorno visual.

1. Border

Que hace: Una propiedad abreviada para establecer el ancho, estilo y color de los bordes de un elemento.

Ejemplo de uso: img { border: 2px solid #ccc; }

Se suele aplicar a: imagenes, div, botones, inputs, tablas, y otros elementos que se benefician de un contorno visual.

1. Border-radius

Que hace: Añade esquinas redondeadas a un elemento.

Ejemplo de uso: button { border-radius: 8px; }

Se suele aplicar a: Botones, imagenes, div, inputs, y otros elementos con bordes para suavizar su apariencia.

1. Width y height

Que hace: Establecen el ancho y alto de un elemento, respectivamente

Ejemplo de uso: div { width: 300´x; height: 200px; }

**Se suele aplicar a:** Imágenes, videos, div, section, y otros contenedores que necesitan dimensiones específicas.

1. Margin

· **Qué hace:** Establece el espacio exterior alrededor de un elemento, separándolo de otros elementos. Es una propiedad abreviada para margin-top, margin-right, margin-bottom, margin-left.

· **Ejemplo de uso:** p { margin: 15px 0; } (15px arriba/abajo, 0 izquierda/derecha)

· **Se suele aplicar a:** Párrafos, encabezados, imágenes, div, y la mayoría de los elementos de bloque para controlar el espaciado.

1. Padding

· **Qué hace:** Establece el espacio interior entre el contenido de un elemento y sus bordes. Es una propiedad abreviada para padding-top, padding-right, padding-bottom, padding-left.

· **Ejemplo de uso:** button { padding: 10px 20px; }

· **Se suele aplicar a:** Botones, div, inputs, encabezados, y cualquier elemento donde se necesite espacio entre el contenido y el borde.

1. Display

· **Qué hace:** Define cómo se muestra un elemento en el diseño. Los valores comunes incluyen block, inline, inline-block, flex, grid, none.

· · **Ejemplo de uso:** li { display: inline-block; } o div { display: flex; }

· · **Se suele aplicar a:** Prácticamente cualquier elemento para cambiar su comportamiento de diseño (por ejemplo, hacer que elementos de lista se muestren en una línea, o usar Flexbox/Grid para la disposición).

1. Float

· **Qué hace:** Coloca un elemento a la izquierda o a la derecha de su contenedor, permitiendo que el contenido fluya alrededor de él.

· · **Ejemplo de uso:** img { float: left; margin-right: 15px; }

· · **Se suele aplicar a:** Imágenes y a veces pequeños bloques de texto o elementos para crear diseños de texto envolvente o columnas simples.

1. Position

· **Qué hace:** Controla la forma en que un elemento se posiciona en el documento. Los valores comunes son static, relative, absolute, fixed, sticky.

· · **Ejemplo de uso:** div.modal { position: fixed; top: 0; left: 0; }

· · **Se suele aplicar a:** Elementos que necesitan ser posicionados de forma precisa, como barras de navegación fijas, modales, o elementos superpuestos.

1. Opacity

· **Qué hace:** Establece el nivel de transparencia de un elemento. Los valores van de 0 (completamente transparente) a 1 (completamente opaco).

· · **Ejemplo de uso:** img.ghost { opacity: 0.5; }

· · **Se suele aplicar a:** Imágenes, contenedores, y cualquier elemento que se desee hacer semitransparente.

1. Box-shadow

· **Qué hace:** Añade una sombra a uno o más lados de un elemento.

· · **Ejemplo de uso:** div.card { box-shadow: 5px 5px 10px rgba(0, 0, 0, 0.2); }

· · **Se suele aplicar a:** Tarjetas, botones, imágenes, y otros elementos para darles profundidad visual.

1. Transition

**Qué hace:** Permite definir un efecto de transición suave al cambiar una propiedad CSS, en lugar de un cambio instantáneo.

**Ejemplo de uso:** button { transition: background-color 0.3s ease-in-out; } (aplicado al estado normal del botón, no al :hover)

**Se suele aplicar a:** Botones, enlaces, imágenes, y elementos interactivos para animar sus cambios de estado (por ejemplo, al pasar el ratón por encima).

1. Z-index

· **Qué hace:** Controla el orden de apilamiento de los elementos posicionados (elementos con position: relative, absolute, fixed, sticky). Los valores más altos aparecen por encima de los valores más bajos.

· · **Ejemplo de uso:** div.modal { z-index: 100; }

· · **Se suele aplicar a:** Modales, menús desplegables, barras de navegación fijas y cualquier elemento que necesite superponerse a otros.

El **DOM (Document Object Model)** es una **interfaz de programación para documentos HTML y XML**. En términos más simples, es una **representación estructurada del documento como un árbol de objetos**, donde cada nodo del árbol es un objeto que representa una parte del documento (un elemento, un atributo, un texto, etc.).

Piensa en una página web como una casa. El HTML es el plano de la casa (dónde están las paredes, las ventanas, las puertas). El CSS es la pintura, los muebles y la decoración que hacen que la casa se vea bonita. El **DOM es como la estructura interna y flexible de la casa** que JavaScript puede entender y modificar.

Cuando un navegador carga una página web, no solo lee el código HTML. En cambio, construye esta representación del DOM en la memoria del ordenador. Esta representación es dinámica, lo que significa que JavaScript puede acceder a ella, cambiar su estructura, estilo y contenido, y el navegador reflejará esos cambios en la pantalla.

**En resumen, el DOM representa:**

**La estructura lógica de la página:** Cómo los elementos están anidados y relacionados entre sí (padres, hijos, hermanos).

**El contenido de la página:** Todo el texto, imágenes, videos, formularios, etc.

**El estilo de la página:** Los estilos CSS aplicados a los elementos.

**Los eventos:** Las interacciones del usuario (clics, movimientos del ratón, entradas de teclado) y cómo la página responde a ellos.7

JavaScript manipula el DOM accediendo a los nodos del árbol y modificando sus propiedades, añadiendo o eliminando nodos, o alterando sus estilos. Esto se hace a través de un conjunto de APIs (Application Programming Interfaces) que el navegador proporciona.