

## Fundamentos de HTML, CSS y Diseño Web

El desarrollo web moderno se cimienta en una comprensión profunda de cómo se estructuran, estilizan y se hacen interactivas las páginas web, así como de la manera en que se adaptan a diversos dispositivos.

### 1. HTML (HyperText Markup Language): La Estructura Semántica de la Web

HTML es la columna vertebral de cualquier página web. No se clasifica como un lenguaje de programación en sentido estricto, sino como un lenguaje de marcado cuyo propósito es estructurar y dotar de significado semántico al contenido. Su funcionamiento se basa en el uso de etiquetas, delimitadas por corchetes angulares (< y >), que definen la función y el tipo de cada elemento, ya sea un párrafo, una imagen, un hipervínculo o una sección específica de contenido.

Los documentos HTML son archivos de texto plano, reconocibles por las extensiones .html o .htm, y son interpretados y visualizados por cualquier navegador web. Históricamente, HTML forma parte del estándar más amplio SGML (Standard Generalized Markup Language), lo cual subraya su rol como un lenguaje para la definición de otros lenguajes de marcado. Su origen se remonta a 1989, cuando Tim Berners-Lee concibió la idea de interconectar documentos a través de hipertexto, sentando así las bases de la World Wide Web.

#### Etiquetas Fundamentales en HTML5:

- Para la construcción de un documento web bien formado y compatible con los estándares actuales, ciertas etiquetas resultan esenciales:
- <!doctype html>: Esta declaración inicial es crucial, ya que instruye al navegador a interpretar el documento según la especificación HTML5, garantizando un renderizado consistente y moderno.
- <html>: Conocido como el "elemento raíz", engloba la totalidad del contenido de la página web. Funciona como el contenedor principal de todos los demás elementos HTML.
- <head>: Contiene metadatos del documento, es decir, información sobre la página que no es directamente visible para el usuario. Aquí se incluyen los enlaces a hojas de estilo, scripts, la especificación de la codificación de caracteres y datos relevantes para el SEO (Search Engine Optimization).
- <title>: Define el título de la página, el cual se exhibe en la pestaña del navegador o en la barra de título de la ventana. Es un elemento fundamental para la identificación del sitio.
- <body>: Este elemento actúa como el contenedor de todo el contenido visible de la página web, abarcando texto, imágenes, videos, enlaces, formularios, entre otros. Todo aquello con lo que el usuario interactúa visualmente se ubica dentro de esta etiqueta.

Etiquetas Semánticas de Sección: HTML5 introdujo un conjunto de etiquetas diseñadas para proporcionar una estructura más significativa y

legible tanto para desarrolladores como para máquinas (motores de búsqueda, lectores de pantalla). Estas incluyen:

- `<section>`: Empleada para agrupar contenido temáticamente relacionado.
- `<nav>`: Contiene los enlaces de navegación principales de la página.
- `<article>`: Destinada a contenido independiente y autocontenido, como una entrada de blog, un artículo de noticias o un comentario.
- `<aside>`: Utilizada para contenido que, aunque relacionado, es secundario al contenido principal (por ejemplo, barras laterales o citas).
- `<header>`: Representa el encabezado de una sección o de la página en su totalidad, a menudo conteniendo un título, un logotipo y elementos de navegación.
- `<footer>`: El pie de una sección o de la página, típicamente con información de derechos de autor, enlaces de contacto o navegación secundaria.
- `<main>`: Representa el contenido principal y único del `<body>` del documento. Solo debe haber un elemento `<main>` por página.
- `<address>`: Proporciona información de contacto para el autor o propietario de un documento o una parte específica de él.
- Etiquetas de Encabezado (`<h1>` a `<h6>`): Definen la jerarquía de los títulos y subtítulos dentro del contenido, donde `<h1>` denota el nivel de importancia más alto y `<h6>` el más bajo. Su correcta aplicación es crucial para la estructura semántica del documento y su accesibilidad.

## **2. DOM (Document Object Model): La Representación Interactiva del Documento**

El Document Object Model (DOM) es una interfaz de programación de aplicaciones (API) fundamental para documentos HTML y XML. Permite que los programas y scripts (especialmente JavaScript) accedan, manipulen y modifiquen el contenido, la estructura y los estilos de las páginas web de manera dinámica.

Conceptualmente, el DOM representa el documento HTML como una estructura de árbol de nodos, donde cada elemento HTML, atributo, segmento de texto, etc., se transforma en un nodo individual. El elemento `<html>` se posiciona como la raíz de este árbol jerárquico. Esta representación facilita la navegación a través del documento y su modificación programática en tiempo real.

Dentro del contexto de la programación web, el objeto global `document` en JavaScript es la puerta de entrada principal al DOM, permitiendo a los desarrolladores interactuar programáticamente con la página cargada en el navegador.

Es importante destacar que el DOM no opera de forma aislada. Se combina con el CSSOM (CSS Object Model), que es una representación similar en árbol de todos los estilos CSS aplicados al documento. La fusión de estos dos modelos da origen al Árbol de Renderizado, una estructura optimizada que contiene exclusivamente los elementos visibles del DOM con sus estilos computados, lista para ser pintada en la pantalla del usuario.

### **3. Ciclo de Vida de una Página Web (Critical Rendering Path)**

El "Critical Rendering Path" describe la secuencia de pasos que sigue un navegador para transformar el código HTML, CSS y JavaScript en píxeles visibles en la pantalla del usuario. Comprender este proceso es esencial para optimizar el rendimiento y la velocidad de carga de las aplicaciones web.

- **Construcción del DOM:** El navegador inicia el proceso parseando el código HTML y construyendo el árbol DOM, que representa la estructura jerárquica del documento.
- **Construcción del CSSOM:** Paralelamente, el navegador parsea el código CSS y construye el árbol CSSOM, que encapsula todas las reglas de estilo aplicadas.
- **Creación del Árbol de Renderizado:** El DOM y el CSSOM se combinan para formar el Árbol de Renderizado. Este árbol incluye únicamente los nodos visibles del DOM y sus estilos calculados, excluyendo, por ejemplo, los elementos que tienen la propiedad `display: none`.
- **Cálculo del Diseño (Layout / Reflow):** Una vez que el Árbol de Renderizado está completo, el navegador calcula la posición y el tamaño exactos de cada elemento dentro de la ventana gráfica. Este proceso determina cómo se distribuirán y ocuparán espacio los elementos en la pantalla.
- **Pintado (Paint / Rasterization):** Finalmente, los píxeles individuales se "pintan" o renderizan en la pantalla, visualizando los elementos con sus colores, fondos, bordes y otras propiedades, según lo definido en la etapa de layout.

### **4. CSS (Cascading Style Sheets): La Estilización y Presentación Visual**

Mientras HTML se encarga de la estructura del contenido, CSS es el lenguaje que dicta la estética y la presentación visual del contenido en la web. Permite a los diseñadores y desarrolladores controlar una vasta gama de aspectos como el color, la tipografía, el espaciado entre elementos, la disposición general y la adaptabilidad a diferentes dispositivos, transformando un documento HTML básico en una experiencia visualmente atractiva y funcional.

Los estilos CSS pueden ser aplicados mediante diversas metodologías:

- **Estilos Internos:** Se insertan directamente en el documento HTML dentro de la etiqueta `<style>`, ubicada comúnmente en la sección `<head>`. Este enfoque es adecuado para estilos muy específicos de una única página.

- **Estilos Externos:** Considerada la práctica más recomendada para proyectos grandes. Los estilos se definen en un archivo .css separado y se enlazan al documento HTML mediante la etiqueta <link> en la sección <head>. Esto fomenta la reutilización del código, la modularidad y simplifica enormemente el mantenimiento de los estilos a lo largo de un sitio.

### Selectores CSS: El Arte de Apuntar

Los selectores CSS son patrones que permiten identificar de manera precisa a qué elementos HTML se aplicarán las reglas de estilo. Dominar el uso de los selectores es fundamental para un control preciso y eficiente del diseño. Incluyen:

- **Selectores de Tipo:** Aplican estilos a todos los elementos de un tipo de etiqueta HTML específico (ej., p para párrafos, h1 para encabezados de nivel 1).
- **Selectores de Clase:** Apuntan a elementos que poseen un atributo class específico (ej., .mi-clase). Permiten aplicar el mismo estilo a múltiples elementos, incluso si no están relacionados semánticamente.
- **Selectores de ID:** Identifican un elemento único en la página mediante su atributo id (ej., #mi-id). Un ID debe ser estrictamente único dentro de un mismo documento HTML.
- **Selectores de Atributo:** Seleccionan elementos basándose en la presencia o el valor de sus atributos (ej., [type="text"] para seleccionar todos los inputs de texto).
- **Combinadores:** Permiten construir selectores más complejos al especificar relaciones entre elementos (ej., descendientes, hijos directos, hermanos adyacentes), brindando un control granular sobre la selección.

### Unidades de Medida en CSS: Flexibilidad y Precisión

CSS ofrece una variedad de unidades de medida para definir tamaños y espaciados, clasificadas en relativas y absolutas, cada una con sus aplicaciones óptimas:

- **Unidades Relativas:** Su valor se calcula en relación con otro elemento o el tamaño de la ventana gráfica, lo que las hace ideales para el diseño responsivo y adaptable. Ejemplos incluyen: em (relativo al tamaño de fuente del elemento padre), rem (relativo al tamaño de fuente del elemento raíz <html>), ch (relativo al ancho del carácter '0'), vw (porcentaje del ancho de la ventana gráfica), vh (porcentaje de la altura de la ventana gráfica) y % (porcentaje del tamaño del elemento padre).
- **Unidades Absolutas:** Poseen un tamaño fijo e inmutable, independientemente del tamaño de la pantalla o del elemento padre. Son útiles en contextos donde se requiere una precisión similar a la de impresión. Ejemplos: px (píxeles), cm (centímetros), mm (milímetros), in (pulgadas), pt (puntos, 1/72 de pulgada), pc (picas, 12 puntos).

## Flexbox (Flexible Box Layout)

Flexbox es un módulo de diseño de CSS que ha revolucionado la forma en que los desarrolladores abordan la disposición unidimensional de elementos en una página web. Permite alinear, distribuir y controlar el orden de los elementos dentro de un contenedor (denominado "contenedor flexible") de manera flexible y eficiente. Es especialmente potente para:

- Centrar elementos de forma sencilla.
- Crear barras de navegación responsivas.
- Distribuir espacios entre elementos de manera uniforme o personalizada.
- Alinear elementos en una sola fila o columna, facilitando la creación de componentes complejos.
- Su capacidad para adaptarse a diferentes tamaños de contenido y contenedores lo convierte en una herramienta indispensable para el diseño responsivo moderno.

## CSS Grid Layout:

CSS Grid Layout es otro módulo de diseño moderno que complementa a Flexbox, enfocándose en la disposición bidimensional. Permite a los desarrolladores crear estructuras de rejilla complejas definiendo explícitamente filas y columnas, y luego colocando elementos en celdas específicas de esa rejilla. Grid es ideal para: