



¡Hola, coders!

Tenemos una misión especial para ustedes: explorar y dominar los fundamentos de CSS a través de exposiciones teóricas y prácticas. Cada equipo abordará un tema esencial que fortalecerá nuestras habilidades en diseño web. A continuación, se detallan las asignaciones:

Alphacoders : Formas de Incorporar CSS en HTML

Objetivo: Investigar y demostrar las tres formas principales de aplicar CSS a un documento HTML:

- **Estilos en línea:** Uso del atributo style directamente en los elementos HTML.
- **Estilos internos:** Uso de una etiqueta <style> dentro del <head> del documento.
- **Estilos externos:** Enlace a un archivo .css externo mediante la etiqueta <link>.

Ejercicio Práctico:

Crear una página HTML simple y aplicar estilos utilizando las tres formas mencionadas para observar las diferencias y ventajas de cada método.

Clean Code: Selectores en CSS

Objetivo: Explorar cómo los selectores permiten aplicar estilos a elementos específicos:

- **Selectores de tipo:** Aplican estilos a todas las instancias de un elemento (e.g., p, h1).
- **Selectores de clase:** Aplican estilos a elementos con un atributo class específico (e.g., .mi-clase).
- **Selectores de ID:** Aplican estilos a un elemento con un atributo id específico (e.g., #mi-id).
- **Selectores de atributo y combinadores:** Permiten una selección más precisa basada en atributos o relaciones entre elementos.

Ejercicio Práctico:



Diseñar una lista de elementos y aplicar diferentes estilos utilizando cada tipo de selector para comprender su alcance y especificidad.

Ctrl Z team: ¿Cómo se carga y procesa el CSS en el navegador?

Objetivo: Comprender el flujo de trabajo del navegador al aplicar estilos CSS

1. El navegador carga el HTML y construye el DOM (Modelo de Objetos del Documento).
2. Busca y carga las hojas de estilo CSS vinculadas.
3. Analiza las reglas CSS y las aplica al DOM para crear el árbol de renderizado.
4. Pinta los estilos en la pantalla.

Ejercicio Práctico:

- Crear una página HTML con una hoja de estilo externa.
- Utilizar las herramientas de desarrollo del navegador (como las DevTools de Chrome) para observar cómo se cargan y aplican los estilos.
- Modificar el CSS en tiempo real y observar los cambios en la página.

Ctrl Z team: Flexbox: Diseño Flexible

Objetivo: Introducir Flexbox como un modelo de diseño unidimensional que facilita la distribución y alineación de elementos dentro de un contenedor.

- **Propiedades del contenedor:** display: flex;, justify-content, align-items.
- **Propiedades de los elementos hijos:** flex-grow, flex-shrink, flex-basis.

Ejercicio Práctico:

Diseñar una barra de navegación horizontal con elementos distribuidos equitativamente utilizando Flexbox.



Mark 1: Grid: Diseño en Dos Dimensiones

Objetivo: Explorar CSS Grid como un sistema de diseño bidimensional que permite crear layouts complejos.

- **Definición de filas y columnas:** grid-template-rows, grid-template-columns.
- **Ubicación de elementos:** grid-row, grid-column, grid-area.

Ejercicio Práctico:

Crear una galería de imágenes organizada en una cuadrícula de 3x3 utilizando CSS Grid.

Wolverine: Introducción a Tailwind CSS

Objetivo: Aprender cómo Tailwind CSS, un framework de CSS utility-first, permite aplicar estilos directamente en el HTML utilizando clases predefinidas.

Ejercicio Práctico:

- Configurar un proyecto básico con Tailwind CSS. Usando CDN
- Recrear un componente común (como una tarjeta de presentación) utilizando las clases utilitarias de Tailwind.
- Comparar el enfoque de Tailwind con el CSS tradicional en términos de velocidad y eficiencia.

Instrucciones Generales:

- Cada equipo deberá escoger uno o máximo dos representantes de cada célula para la exposición 5 a 10 minutos max para su exposición.
- Cada equipo tendrá 5 a 10 minutos max para su exposición.
- La presentación debe incluir una explicación teórica clara y una demostración práctica.
- Se valorará la creatividad, claridad y aplicabilidad de los ejemplos.