**Clase ¡Tu primera página web con HTML y CSS!**

**Objetivo de aprendizaje:**

Comprender los elementos básicos de HTML y CSS para crear la estructura y el estilo de una página web sencilla.

**Introducción**

¿Sabías que toda página web comienza con unas simples líneas de código? En esta clase vas a descubrir cómo escribir tu primer documento HTML y darle estilo con CSS. No necesitas ser experto, solo tener curiosidad y ganas de aprender.

**Desarrollo paso a paso**

**1. ¿Qué es HTML?**

HTML significa *HyperText Markup Language*. Es el lenguaje que usamos para estructurar el contenido de una página. No es un lenguaje de programación, sino de marcado.

**Estructura básica:**

html

<!DOCTYPE html>

<html>

<head>

<title>Mi Primera Página</title>

</head>

<body>

<h1>¡Hola, mundo!</h1>

<p>Esta es mi primera página web con HTML.</p>

</body>

</html>

**Explicación:**

* <!DOCTYPE html>: Le dice al navegador que este es un documento HTML5.
* <html>: Contenedor de todo el contenido HTML.
* <head>: Aquí va información "invisible" como el título o enlaces a estilos.
* <title>: Título que aparece en la pestaña del navegador.
* <body>: Todo lo que el usuario ve en pantalla.

**2. ¿Qué es CSS?**

CSS significa *Cascading Style Sheets* y nos permite darle diseño al HTML (colores, fuentes, espacios, etc.).

**Ejemplo sencillo:**

html

<style>

h1 {

color: blue;

font-family: Arial, sans-serif;

}

p {

font-size: 16px;

color: gray;

}

</style>

Esto puede ir dentro de <head> o en un archivo separado.

**Actividad1 práctica creativa**

Crea una página HTML con lo siguiente:

1. Un título principal con tu nombre.
2. Un párrafo con una descripción breve de ti.
3. Usa CSS para:
   * Cambiar el color del título.
   * Cambiar el tamaño del texto del párrafo.
   * Poner un fondo de color claro al body.

**Clase : Listas, tablas y formularios en HTML**

**Objetivo de aprendizaje:**

Crear y usar listas, tablas y formularios en HTML para organizar y recopilar información de los usuarios.

**Introducción**

En una web, no todo es solo texto. Muchas veces necesitamos mostrar datos ordenados o permitir que los usuarios escriban información. Para eso usamos **listas, tablas y formularios**. ¡Hoy aprenderás a dominarlos!

**Desarrollo paso a paso**

**1. Listas en HTML**

Hay dos tipos principales:

* **Listas ordenadas** (<ol>): tienen números.
* **Listas desordenadas** (<ul>): tienen puntos.

html

<p>Lenguajes Web:</p>

<ul>

<li>HTML</li>

<li>CSS</li>

<li>JavaScript</li>

</ul>

<p>Pasos para crear una web:</p>

<ol>

<li>Escribir HTML</li>

<li>Diseñar con CSS</li>

<li>Programar con JS</li>

</ol>

**2. Tablas en HTML**

Útiles para mostrar datos de forma clara.

html

<table border="1">

<tr>

<th>Nombre</th>

<th>Edad</th>

</tr>

<tr>

<td>Laura</td>

<td>25</td>

</tr>

<tr>

<td>José</td>

<td>30</td>

</tr>

</table>

**Etiquetas clave:**

* <table>: comienza la tabla.
* <tr>: fila.
* <th>: encabezado.
* <td>: celda.

**3. Formularios en HTML**

Capturan datos del usuario.

html

<form>

<label for="nombre">Nombre:</label>

<input type="text" id="nombre" name="nombre"><br><br>

<label for="email">Correo:</label>

<input type="email" id="email" name="email"><br><br>

<input type="submit" value="Enviar">

</form>

**Otros campos comunes:**

* <textarea>: área de texto.
* <select> y <option>: lista desplegable.
* type="radio" o type="checkbox" para opciones.

**Actividad2 práctica creativa**

Crea una página HTML que contenga:

1. Una lista ordenada con tus 3 metas de este año.
2. Una tabla con tus 3 materias favoritas y la calificación que esperas obtener.
3. Un formulario con:
   * Nombre
   * Correo electrónico
   * Una pregunta: "¿Qué esperas aprender en este curso?"

**Clase : Embellece tu web con CSS: listas, tablas y formularios**

**Objetivo de aprendizaje:**

Aplicar estilos con CSS para mejorar la presentación de listas, tablas y formularios en una página HTML.

**Introducción**

Hasta ahora, nuestras listas, tablas y formularios eran funcionales… ¡pero bastante aburridos! Hoy los haremos más atractivos visualmente con CSS, usando colores, espaciado, bordes y tipografías.

**Desarrollo paso a paso con ejemplos**

**1. Estilizando listas**

html

<ul class="lista-elegante">

<li>HTML</li>

<li>CSS</li>

<li>JavaScript</li>

</ul>

<style>

.lista-elegante {

list-style-type: square;

color: darkblue;

font-family: Arial;

background-color: #f0f8ff;

padding: 10px;

border-radius: 8px;

}

</style>

**¿Qué hicimos aquí?**

* list-style-type: cambia el estilo de la viñeta.
* background-color: pone un fondo bonito.
* padding y border-radius: suavizan el contenedor.

**2. Estilizando tablas**

html

<table class="tabla-colorida">

<tr>

<th>Nombre</th>

<th>Edad</th>

</tr>

<tr>

<td>Laura</td>

<td>25</td>

</tr>

<tr>

<td>José</td>

<td>30</td>

</tr>

</table>

<style>

.tabla-colorida {

width: 100%;

border-collapse: collapse;

}

.tabla-colorida th, .tabla-colorida td {

border: 1px solid #999;

padding: 8px;

text-align: left;

}

.tabla-colorida tr:nth-child(even) {

background-color: #f9f9f9;

}

.tabla-colorida th {

background-color: #4CAF50;

color: white;

}

</style>

**Trucos usados:**

* border-collapse: une los bordes.
* nth-child(even): aplica estilo a filas pares.
* Colores para distinguir encabezados y filas.

**3. Estilizando formularios**

html

<form class="formulario">

<label for="nombre">Nombre:</label>

<input type="text" id="nombre"><br><br>

<label for="email">Correo:</label>

<input type="email" id="email"><br><br>

<input type="submit" value="Enviar">

</form>

<style>

.formulario {

background-color: #fff8dc;

padding: 20px;

width: 300px;

border: 2px solid #ccc;

border-radius: 10px;

font-family: Verdana;

}

.formulario label {

display: block;

margin-bottom: 5px;

}

.formulario input[type="text"],

.formulario input[type="email"] {

width: 100%;

padding: 5px;

margin-bottom: 10px;

}

.formulario input[type="submit"] {

background-color: #4CAF50;

color: white;

border: none;

padding: 10px;

width: 100%;

cursor: pointer;

}

.formulario input[type="submit"]:hover {

background-color: #45a049;

}

</style>

**Estilos clave:**

* width: 100%: hace que los inputs ocupen el ancho total.
* hover: cambia el color al pasar el cursor.

**Actividad3 práctica creativa**

1. Usa la actividad anterior (listas, tabla y formulario).
2. Aplica estilos CSS para:
   * Poner un fondo a la tabla.
   * Cambiar la fuente de la lista.
   * Colorear el botón del formulario.
3. Guarda el archivo como estilos.html y ábrelo en tu navegador.

**Clase: ¡Diseño flexible con Flexbox!**

**Objetivo de aprendizaje:**

Comprender los conceptos básicos de Flexbox y aplicarlos para alinear y distribuir elementos en una página web.

**Introducción**

¿Te ha pasado que colocas dos cajas en HTML y no sabes cómo alinearlas bien? ¿O quieres que se centren en la pantalla y no sabes cómo? Flexbox llegó para solucionar eso. Es como decirle a CSS: “¡organiza esto por mí!”

**Desarrollo paso a paso con ejemplos**

**1. Activar Flexbox**

html

<div class="contenedor">

<div class="caja">A</div>

<div class="caja">B</div>

<div class="caja">C</div>

</div>

<style>

.contenedor {

display: flex;

background-color: #eee;

padding: 10px;

}

.caja {

background-color: #4CAF50;

color: white;

padding: 20px;

margin: 5px;

text-align: center;

width: 100px;

}

</style>

**¿Qué hicimos?**

* display: flex; transforma el contenedor en un “contenedor flexible”.
* Las .caja se alinean horizontalmente automáticamente.

**2. Centrar elementos**

css

<!DOCTYPE html>

<html>

<head>

<title>Ejemplo Flexbox</title>

<style>

.contenedor {

display: flex;

justify-content: center; /\* Alinea horizontalmente \*/

align-items: center; /\* Alinea verticalmente \*/

height: 300px;

background-color: #ddd;

}

.caja {

background-color: #4CAF50;

color: white;

padding: 20px;

margin: 10px;

}

</style>

</head>

<body>

<div class="contenedor">

<div class="caja">Caja 1</div>

<div class="caja">Caja 2</div>

<div class="caja">Caja 3</div>

</div>

</body>

</html>

**3. Distribuir el espacio**

css

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>Ejemplo de justify-content en Flexbox</title>

<style>

body {

font-family: Arial, sans-serif;

padding: 20px;

}

h2 {

margin-top: 40px;

}

.contenedor {

display: flex;

align-items: center;

height: 100px;

background-color: #ddd;

margin-bottom: 30px;

border: 2px solid #aaa;

}

.caja {

background-color: #4CAF50;

color: white;

padding: 20px;

margin: 5px;

text-align: center;

width: 100px;

}

.between {

justify-content: space-between;

}

.around {

justify-content: space-around;

}

.evenly {

justify-content: space-evenly;

}

</style>

</head>

<body>

<h1>Demostración de justify-content en Flexbox</h1>

<h2>space-between</h2>

<div class="contenedor between">

<div class="caja">Caja 1</div>

<div class="caja">Caja 2</div>

<div class="caja">Caja 3</div>

</div>

<h2>space-around</h2>

<div class="contenedor around">

<div class="caja">Caja 1</div>

<div class="caja">Caja 2</div>

<div class="caja">Caja 3</div>

</div>

<h2>space-evenly</h2>

<div class="contenedor evenly">

<div class="caja">Caja 1</div>

<div class="caja">Caja 2</div>

<div class="caja">Caja 3</div>

</div>

</body>

</html>**4. Cambiar la dirección**

css

flex-direction: column; /\* De vertical en lugar de horizontal \*/

**Actividad4 práctica creativa**

Crea una página con un contenedor que tenga 3 cajas:

* Usa Flexbox para centrar las cajas horizontal y verticalmente.
* Luego cambia la dirección a column.
* Prueba diferentes valores en justify-content.

Guarda el archivo como flexbox.html y experimenta con los valores.

**Clase : Crea tu portafolio personal con Flexbox**

**Objetivo de aprendizaje:**

Diseñar la estructura completa de una página tipo CV/portafolio utilizando Flexbox para organizar cabecera, contenido principal y pie de página.

**Introducción**

¿Te imaginas tener tu propio sitio web donde puedas mostrar tus habilidades, estudios y experiencia? Con Flexbox podemos dividir fácilmente una página en secciones claras y ordenadas. Hoy construiremos esa estructura base.

**Desarrollo paso a paso con ejemplo**

**Estructura general del layout:**

html

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>Mi Portafolio</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

}

.layout {

display: flex;

flex-direction: column;

min-height: 100vh;

}

header {

background-color: #333;

color: white;

padding: 20px;

text-align: center;

}

main {

flex: 1;

display: flex;

flex-direction: row;

padding: 20px;

background-color: #f4f4f4;

}

.sidebar {

width: 30%;

background-color: #eee;

padding: 15px;

margin-right: 20px;

}

.contenido {

flex: 1;

background-color: white;

padding: 15px;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px;

}

</style>

</head>

<body>

<div class="layout">

<header>

<h1>Juan Pérez - Desarrollador Web</h1>

</header>

<main>

<div class="sidebar">

<h3>Contacto</h3>

<p>Email: juan@example.com</p>

<p>Tel: 123456789</p>

<h3>Habilidades</h3>

<ul>

<li>HTML</li>

<li>CSS</li>

<li>JavaScript</li>

</ul>

</div>

<div class="contenido">

<h2>Sobre mí</h2>

<p>Soy desarrollador web con pasión por crear sitios atractivos y funcionales.</p>

<h2>Experiencia</h2>

<p><strong>2023 - Actual</strong> - Desarrollador Front-End en WebCorp</p>

<h2>Proyectos destacados</h2>

<ul>

<li>Mi tienda online</li>

<li>App de clima con API</li>

</ul>

</div>

</main>

<footer>

<p>© 2025 Juan Pérez</p>

</footer>

</div>

</body>

</html>

**Actividad6 práctica creativa**

1. Personaliza el ejemplo con tu nombre, experiencia y datos reales (o inventados).
2. Cambia los colores y fuentes con CSS para que reflejen tu estilo.
3. Agrega una imagen tuya en la cabecera o sidebar usando <img>.

**Ejercicio1**

**Vamos a desglosar línea por línea el archivo index.html del CV web para que entiendas perfectamente qué hace cada parte:**

**html**

**<!DOCTYPE html>**

**Le dice al navegador que este documento está escrito en HTML5 (versión actual de HTML).**

**<html lang="es">**

**Abre el documento HTML y especifica que el idioma principal del contenido es español (lang="es").**

**<head>**

**Comienza la sección <head>, que contiene información sobre el documento que no se muestra directamente en la página, pero que es importante para el navegador.**

**<meta charset="UTF-8">**

**Indica que se usarán caracteres UTF-8, lo que permite mostrar acentos, eñes, símbolos, etc.**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**Hace que la página se vea bien en celulares y tablets. Es clave para el diseño responsivo.**

**<title>CV - Juan Pérez</title>**

**Define el título de la pestaña del navegador.**

**<link rel="stylesheet" href="styles.css">**

**Conecta este archivo HTML con el archivo de estilos styles.css.**

**</head>**

**<body>**

**Aquí termina el <head> y empieza el <body>, donde va todo el contenido que sí se ve en pantalla.**

**<header>**

**Comienza el encabezado del CV, donde pondremos la foto y el nombre.**

**<img src="https://via.placeholder.com/120" alt="Foto de perfil" class="foto-perfil">**

**Muestra una imagen de perfil con estilo (class="foto-perfil").**

* **src es la URL de la imagen.**
* **alt es un texto alternativo que aparece si la imagen no se carga.**

**<h1>Juan Pérez</h1>**

**Es el nombre grande que se verá en la parte superior. Un <h1> es el encabezado principal de la página.**

**</header>**

**Cierra el encabezado.**

**<section class="contacto">**

**Comienza una sección para los datos de contacto, con la clase contacto (usada en CSS si quieres personalizar estilos).**

**<h2>Contacto</h2>**

**Título de segundo nivel para esta sección.**

**<p>Email: juanperez@example.com</p>**

**<p>Teléfono: 123-456-7890</p>**

**<p>Ciudad: Ciudad de México</p>**

**Tres párrafos que muestran el correo, teléfono y ciudad.**

**</section>**

**Cierra la sección de contacto.**

**<section class="resumen">**

**<h2>Resumen</h2>**

**<p>...</p>**

**</section>**

**Sección donde se describe brevemente tu perfil o experiencia. Ideal para destacar tus puntos fuertes.**

**<section class="habilidades">**

**<h2>Habilidades</h2>**

**<ul>**

**<li>HTML5 y CSS3</li>**

**<li>JavaScript y React</li>**

**<li>Diseño responsivo</li>**

**<li>Git y GitHub</li>**

**</ul>**

**</section>**

**Lista de habilidades técnicas usando:**

* **<ul> para una lista sin orden.**
* **<li> para cada habilidad.**

**<section class="experiencia">**

**<h2>Experiencia Laboral</h2>**

**<p><strong>Desarrollador Front-End</strong> - WebTech Solutions (2022 - Actual)</p>**

**<p>Diseño y desarrollo de sitios web responsivos y accesibles.</p>**

**<p><strong>Diseñador Web Junior</strong> - Creativa Studio (2020 - 2022)</p>**

**<p>Apoyo en maquetación de sitios, mejora de estilos y prototipado.</p>**

**</section>**

**Sección con tus trabajos anteriores:**

* **<strong> resalta el cargo.**
* **Cada <p> representa una experiencia o descripción.**

**</body>**

**</html>**

**Cierran el cuerpo de la página y el documento HTML.**

**<!-- Esto es un comentario en HTML -->**

Vamos a analizar **línea por línea** el archivo styles.css del Currículum Vítae Web. Esto te ayudará a comprender cómo se aplica el diseño visual al HTML.

css

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f9f9f9;

color: #333;

}

Aplica estilos generales al **cuerpo de la página**:

* font-family: define la fuente del texto.
* margin y padding: eliminan los espacios por defecto.
* background-color: color de fondo claro para toda la página.
* color: color del texto (gris oscuro).

header {

background-color: #4CAF50;

color: white;

text-align: center;

padding: 40px 20px;

}

Estilo del encabezado:

* background-color: verde fuerte.
* color: texto en blanco.
* text-align: centra el texto.
* padding: espacio interno arriba y abajo (40px) y a los lados (20px).

.foto-perfil {

border-radius: 50%;

width: 120px;

height: 120px;

border: 4px solid white;

margin-bottom: 15px;

}

Estilo de la **foto de perfil**:

* border-radius: 50%: la hace circular.
* width y height: tamaño de 120px.
* border: borde blanco de 4px.
* margin-bottom: espacio debajo de la foto.

section {

max-width: 700px;

margin: 30px auto;

padding: 20px;

background-color: white;

border-radius: 10px;

}

Estilo general para **cada sección del CV**:

* max-width: no se hace más ancha de 700px.
* margin: 30px auto: separa verticalmente y centra horizontalmente.
* padding: espacio interno para que el contenido no quede pegado al borde.
* background-color: fondo blanco para destacar sobre el gris claro del body.
* border-radius: esquinas redondeadas suaves.

h2 {

color: #4CAF50;

margin-bottom: 10px;

}

Estilo para los subtítulos (<h2>):

* color: verde igual al encabezado.
* margin-bottom: separa el título del texto siguiente.

ul {

padding-left: 20px;

}

Aumenta el espacio al inicio de listas:

* padding-left: separa la lista del borde izquierdo.

ul li {

margin-bottom: 5px;

}

Aplica espacio entre los elementos de la lista:

* margin-bottom: separa visualmente cada ítem.

Con estos estilos conseguimos un diseño **limpio, moderno y legible** que funciona bien tanto en escritorio como en móviles básicos.