MINI TUTORIAL FLEXBOX CSS

1. ¿Qué es Flexbox?

Flexbox es un modelo de diseño unidimensional en CSS que permite **colocar elementos horizontal o verticalmente**, distribuir espacio entre ellos y alinearlos con precisión, independientemente del tamaño de pantalla.

---

2. Activar Flexbox

css

```
.contenedor {
  display: flex;  /* o inline-flex */
}
```

Esto convierte a `.contenedor` en un **contenedor flexible**, y sus hijos se convierten en **ítems flexibles**.

---

3. Propiedades del CONTENEDOR FLEXIBLE

Propiedad	Valores	Efecto
`flex-direction`	row (por defecto), column	Dirección del eje principal
`justify-content`	center, space-between, etc.	Alineación horizontal (eje principal)
`align-items`	center, stretch, etc.	Alineación vertical (eje cruzado)
`align-content`	center, space-between	Alineación de múltiples líneas (cuando hay `wrap`)
`flex-wrap`	nowrap, wrap, wrap-reverse	Permite que los ítems salten de línea
`gap`	px, rem	Espacio entre ítems (nativo desde CSS3)

MINI TUTORIAL FLEXBOX CSS

4. Propiedades de los ÍTEMS FLEXIBLES

Propiedad	Qué hace
`order`	Cambia el orden visual del ítem
`flex-grow`	Cuánto puede crecer en proporción
`flex-shrink`	Cuánto puede encogerse en proporción
`flex-basis`	Tamaño inicial antes de crecer o encoger
`flex`	Atajo para `grow shrink basis` (ej: `flex: 1 1 200px`)
`align-self`	Alineación individual diferente al resto

5. Ejemplos Visuales por Nivel

🔰 Nivel 1: Distribuir horizontalmente

html

```
<div class="contenedor">
  <div class="item">A</div>
  <div class="item">B</div>
  <div class="item">C</div>
</div>
```

css

```
.contenedor {
  display: flex;
  justify-content: space-between;
  background: #eee;
  padding: 1rem;
}
.item {
  background: tomato;
  padding: 1rem;
  color: white;
}
```

MINI TUTORIAL FLEXBOX CSS

Nivel 2: Vertical con `column`

css

```css
.contenedor {
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

Nivel 3: Cards responsivas

css

```css
.catalogo {
  display: flex;
  flex-wrap: wrap;
  gap: 1rem;
}

.card {
  flex: 1 1 200px;
  background: #fff;
  padding: 1rem;
  border-radius: 8px;
  box-shadow: 0 0 8px rgba(0,0,0,0.1);
}
```

MINI TUTORIAL FLEXBOX CSS

Nivel 4: Menú adaptable

css

```
nav ul {
  display: flex;
  justify-content: space-around;
  background-color: #222;
  list-style: none;
}
nav li a {
  color: white;
  padding: 1rem;
  display: block;
  text-decoration: none;
}
```

---

6. Buenas prácticas

- ✅ Usar `gap` en lugar de `margin` entre ítems
- ✅ Usar `flex-wrap: wrap` para diseños fluidos
- ❌ No usar `float` ni `position` para lo que Flexbox resuelve mejor
- ✅ Usar `flex: 1` para distribuir espacio automáticamente
- ✅ Combinable con Grid para estructuras más complejas

MINI TUTORIAL FLEXBOX CSS

7. Mini Proyecto: Catálogo de Videojuegos con Flexbox

📦 HTML

html

```
<div class="catalogo">
  <div class="juego">
    <img src="eldenring.jpg" alt="">
    <h3>Elden Ring</h3>
    <button>Comprar</button>
  </div>
  <!-- Repetir -->
</div>
```

MINI TUTORIAL FLEXBOX CSS

🎨 CSS

CSS

```css
.catalogo {
  display: flex;
  flex-wrap: wrap;
  gap: 1rem;
  justify-content: center;
}

.juego {
  flex: 1 1 200px;
  background: #fff;
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0,0,0,0.1);
  padding: 1rem;
  text-align: center;
}

.juego img {
  width: 100%;
  border-radius: 6px;
}

.juego h3 {
  margin: 1rem 0 0.5rem;
}

.juego button {
  padding: 0.5rem 1rem;
  background: #4b6cb7;
  color: white;
  border: none;
  border-radius: 4px;
}
```

MINI TUTORIAL FLEXBOX CSS

8. Herramientas para practicar Flexbox

- 🔗 Flexbox Froggy (juego)
- 🔗 CSS Tricks - Guía Visual de Flexbox
- 🔗 Flexbox Defense

MINI TUTORIAL FLEXBOX CSS

1. Modo Oscuro con Flexbox Adaptable

✅ Objetivo:

Aplicar un sistema de **modo claro/oscuro** que funcione con diseño en **Flexbox responsivo**.

🧱 Estructura base HTML

html

```
<body>
  <div class="contenedor">
    <header>
      <h1>Catálogo Gamer</h1>
      <button id="modo-btn">🌓 Modo Oscuro</button>
    </header>

    <main class="catalogo">
      <div class="juego">🎮 Juego 1</div>
      <div class="juego">🎮 Juego 2</div>
      <div class="juego">🎮 Juego 3</div>
    </main>
  </div>
</body>
```

CSS con variables

css

```
:root {
  --fondo: #f5f5f5;
  --texto: #111;
  --carta: #ffffff;
}

[data-tema="oscuro"] {
  --fondo: #111;
  --texto: #f5f5f5;
  --carta: #1e1e1e;
```

MINI TUTORIAL FLEXBOX CSS

```
}

body {
  background: var(--fondo);
  color: var(--texto);
  font-family: sans-serif;
  transition: all 0.3s ease;
}

.contenedor {
  display: flex;
  flex-direction: column;
  align-items: center;
  gap: 2rem;
  padding: 2rem;
}

.catalogo {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  gap: 1rem;
}

.juego {
  background: var(--carta);
  padding: 2rem;
  border-radius: 8px;
  min-width: 180px;
  text-align: center;
  transition: background 0.3s;
}

#modo-btn {
  padding: 0.5rem 1rem;
  background: orange;
  border: none;
  cursor: pointer;
  border-radius: 5px;
```

MINI TUTORIAL FLEXBOX CSS

```
}
```

JavaScript para cambiar modo

html

```
<script>
  const btn = document.getElementById("modo-btn");
  btn.addEventListener("click", () => {
    const tema = document.documentElement.getAttribute("data-tema");
    document.documentElement.setAttribute(
      "data-tema",
      tema === "oscuro" ? "claro" : "oscuro"
    );
  });
</script>
```

---

2. Layouts Dinámicos para Admin o eCommerce con Flexbox

Objetivo:

Crear un diseño de tipo **panel de administración o tienda**, con **barra lateral**, contenido principal y **Flexbox**.

HTML

html

```
<div class="admin-layout">
  <aside class="sidebar">Menú</aside>
  <main class="contenido">
    <header class="topbar">Panel de control</header>
    <section class="panel">Contenido dinámico</section>
  </main>
</div>
```

MINI TUTORIAL FLEXBOX CSS

CSS

css

```css
.admin-layout {
  display: flex;
  min-height: 100vh;
}

.sidebar {
  width: 220px;
  background: #222;
  color: white;
  padding: 1rem;
}

.contenido {
  flex: 1;
  display: flex;
  flex-direction: column;
}

.topbar {
  background: #444;
  color: white;
  padding: 1rem;
}

.panel {
  flex: 1;
  padding: 2rem;
  background: #f9f9f9;
}
```

MINI TUTORIAL FLEXBOX CSS

3. Responsive con Media Queries + Flexbox

🎯 Objetivo:

Adaptar el layout para móviles.

css

```
@media (max-width: 768px) {
  .admin-layout {
    flex-direction: column;
  }

  .sidebar {
    width: 100%;
    order: 2;
  }

  .topbar {
    order: 1;
  }
}
```

✅ Resultado:

- Layout de panel flexible (sidebar + contenido)
- Se reorganiza en móviles (sidebar abajo, barra arriba)
- Compatible con modo oscuro si lo integras
