Uso de CSS Avanzado: Flexbox y Grid

1. Introducción a la maquetación con CSS

Antes de entrar en detalles sobre **Flexbox** y **Grid**, es importante entender el papel de CSS en la estructura de una página web. CSS permite dar **forma, tamaño y distribución** a los elementos dentro de un sitio. Dos de las herramientas más poderosas para lograr diseños eficientes y responsivos son **Flexbox** y **CSS Grid**.

¿Por qué usar Flexbox y Grid?

✓ Facilitan la alineación y distribución de elementos sin necesidad de usar floats o posiciones absolutas.
✓ Hacen que el diseño sea más flexible y adaptable a diferentes tamaños de pantalla.
✓ Reducen la cantidad de código necesario en comparación con técnicas antiguas de maquetación.

---

2. Uso de Flexbox en la maquetación

¿Qué es Flexbox?

El modelo de **Flexbox (Flexible Box Model)** es un sistema de diseño unidimensional que permite organizar elementos de manera eficiente dentro de un contenedor flexible. Se usa principalmente para **alinear elementos en fila o columna** de forma automática.

Propiedades principales de Flexbox

Para usar Flexbox, primero se define un contenedor como "flexible":

Las propiedades más importantes son:

📌 **Dirección de los elementos (flex-direction)**

Define si los elementos estarán organizados en fila o columna.

```
.container {
  display: flex;
  flex-direction: row; /* Por defecto: los elementos se alinean en fila */
}
```

Opciones:

- row (fila, por defecto)
- row-reverse (fila inversa)
- column (columna)
- column-reverse (columna inversa)

📌 **Espaciado entre los elementos (justify-content)**

Permite distribuir los elementos dentro del contenedor de manera equitativa.

```
.container {
  justify-content: space-between; /* Espacio uniforme entre los elementos */
}
```

Opciones:

- flex-start (alineado a la izquierda)
- flex-end (alineado a la derecha)
- center (centrado)
- space-between (espacio uniforme entre elementos)
- space-around (espacios alrededor de cada elemento)

📌 **Alineación vertical (align-items)**

Ajusta cómo se alinean los elementos dentro del contenedor.

```
.container {
  align-items: center; /* Alineación vertical al centro */
}
```

Opciones:

- stretch (por defecto, los elementos se expanden para llenar el contenedor)
- flex-start (alineado en la parte superior)
- flex-end (alineado en la parte inferior)
- center (centrado)

---

3. Uso de CSS Grid en la maquetación

¿Qué es CSS Grid?

CSS **Grid Layout** es un sistema de diseño bidimensional que permite organizar elementos tanto en **filas como en columnas**. Es ideal para la distribución de elementos en una página web de manera más compleja que Flexbox.

Cómo definir un contenedor Grid

Para empezar, se debe declarar el contenedor como display: grid;

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr); /* Crea 3 columnas del mismo tamaño */
  gap: 10px; /* Espaciado entre elementos */
}
```

📌 **Definir columnas y filas**

grid-template-columns: repeat(3, 1fr);

- repeat(3, 1fr): Define **3 columnas** de tamaño igual (1 fracción cada una).

grid-template-rows: 100px 200px auto;

- Define filas con alturas específicas.

📌 **Distribución automática**

grid-auto-rows: minmax(100px, auto);

- Ajusta las filas automáticamente según el contenido.

---

4. Diferencias entre Flexbox y Grid

Característica	Flexbox	Grid
Dimensión	Unidimensional (fila o columna)	Bidimensional (filas y columnas)
Alineación	Mejor para distribuir elementos en una sola dirección	Mejor para diseños complejos en dos direcciones
Ejemplo de uso	Barra de navegación, alineación de elementos	Galerías de productos, maquetación de páginas

---

5. Definir variables CSS y aplicar estilos reutilizables

Para mejorar la organización de los estilos, podemos usar **variables CSS**:

```
:root {
  --primary-color: #007bff;
  --secondary-color: #f8f9fa;
  --font-size: 16px;
}
```

```
body {
  background-color: var(--secondary-color);
  font-size: var(--font-size);
}

.button {
  background-color: var(--primary-color);
  color: white;
  padding: 10px 20px;
  border-radius: 5px;
}
```

✅ **Ventajas de usar variables CSS:**

- Facilita la modificación de estilos en todo el sitio.
- Permite personalizar temas rápidamente.
- Mejora la mantenibilidad del código.

Crear un repositorio en GitHub

1. Ve a **GitHub** e inicia sesión.
2. Haz clic en el botón **New** (Nuevo) o ve a https://github.com/new.
3. Escribe un nombre para el repositorio, por ejemplo: **carrito-compras**.
4. Puedes agregar una descripción opcional.
5. Elige si quieres que el repositorio sea **público** (cualquiera puede verlo) o **privado** (solo tú puedes verlo).
6. No marques la opción de inicializar con README (lo agregaremos manualmente).
7. Haz clic en **Create repository**.

Configurar Git en tu computadora

Si aún no tienes Git instalado, descárgalo aquí:
🔗 https://git-scm.com/downloads

Luego, sigue estos pasos en la terminal (Windows, macOS o Linux):

Configurar tu usuario de Git (solo la primera vez):

git config --global user.name "TuNombre"

git config --global user.email "TuCorreo@example.com"

Verificar que Git está instalado correctamente:

git --version

Crear el proyecto en tu computadora

Abre la terminal o línea de comandos.

Ve a la carpeta donde quieres guardar el proyecto:

cd Desktop # O la carpeta donde quieras guardar el proyecto

Crea una nueva carpeta y entra en ella:

mkdir carrito-compras

cd carrito-compras

Inicializa un repositorio Git en la carpeta:

git init

Agregar los archivos del carrito de compras

Crea los archivos del proyecto:

- **index.html** → Estructura del sitio web.
- **styles.css** → Estilos con Flexbox y Grid.
- **script.js** → Lógica del carrito.

Subir el proyecto a GitHub

Agrega todos los archivos al repositorio Git:

git add .

- Esto añade todos los archivos al área de preparación.

Crea un commit con un mensaje descriptivo:

git commit -m "Primer commit: carrito de compras básico"

Conectar el repositorio local con el repositorio en GitHub
(Copia la URL de tu repositorio desde GitHub, que se verá como https://github.com/tuusuario/carrito-compras.git)

git remote add origin https://github.com/tuusuario/carrito-compras.git

Subir el código al repositorio en GitHub:

git branch -M main

git push -u origin main

Verifica en GitHub

- Abre **GitHub** y entra a tu repositorio.
- Deberías ver los archivos index.html, styles.css y script.js subidos.
- Si hiciste todo correctamente, ¡el carrito de compras ya está en GitHub! 🚀 🎉