**6. LAYOUT CON FLEXBOX**

Flexbox, o Flexible Box Layout, es un modelo de diseño en CSS que facilita la creación de layouts flexibles y eficientes. Flexbox es ideal para distribuir espacio entre elementos y alinear elementos en contenedores, incluso cuando su tamaño es desconocido o dinámico.

**1. Conceptos Básicos de Flexbox**

* **Contenedor Flex**: El elemento padre que contiene los elementos flexibles. Se define con display: flex.
* **Elementos Flex**: Los elementos hijos dentro del contenedor flex.

**2. Propiedades del Contenedor Flex**

* **display**: Activa el modo flex en el contenedor.

**CSS**

.container {

display: flex;

}

* **flex-direction**: Define la dirección de los elementos flexibles (fila o columna).

**CSS**

.container {

flex-direction: row; /\* fila (por defecto) \*/

/\* flex-direction: column; columna \*/

}

* **justify-content**: Alinea los elementos a lo largo del eje principal.

**CSS**

.container {

justify-content: center; /\* Centra los elementos horizontalmente \*/

/\* Otros valores: flex-start, flex-end, space-between, space-around \*/

}

* **align-items**: Alinea los elementos a lo largo del eje transversal.

**CSS**

.container {

align-items: center; /\* Centra los elementos verticalmente \*/

/\* Otros valores: flex-start, flex-end, stretch, baseline \*/

}

**3. Propiedades de los Elementos Flex**

* **flex-grow**: Define cómo un elemento puede crecer en relación con los demás.

**CSS**

.item {

flex-grow: 1; /\* El elemento crecerá para llenar el espacio disponible \*/

}

* **flex-shrink**: Define cómo un elemento puede encogerse en relación con los demás.

**CSS**

.item {

flex-shrink: 1; /\* El elemento puede encogerse si es necesario \*/

}

* **flex-basis**: Define el tamaño inicial de un elemento antes de que se distribuya el espacio restante.

**CSS**

.item {

flex-basis: 200px; /\* Tamaño inicial del elemento \*/

}

**4. Ejemplo Completo de Layout con Flexbox**

**CSS**

/\* Contenedor Flex \*/

.container {

display: flex;

flex-direction: row;

justify-content: space-around;

align-items: center;

height: 100vh; /\* Altura del contenedor \*/

background-color: #f0f0f0;

}

/\* Elementos Flex \*/

.item {

background-color: #3498db;

color: white;

padding: 20px;

margin: 10px;

text-align: center;

flex-grow: 1; /\* Cada elemento crecerá para llenar el espacio disponible \*/

}

**HTML**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

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

<title>Layout con Flexbox</title>

</head>b

<body>

<div class="container">

<div class="item">Elemento 1</div>

<div class="item">Elemento 2</div>

<div class="item">Elemento 3</div>

</div>

</body>

</html>

**Explicación del Ejemplo**

En este ejemplo, el contenedor .container utiliza display: flex para activar el modo Flexbox. Los elementos hijos .item se distribuyen en una fila (flex-direction: row) y se alinean horizontalmente con justify-content: space-around, lo que coloca espacio igual entre ellos. La propiedad align-items: center centra los elementos verticalmente dentro del contenedor.

Cada .item tiene flex-grow: 1, lo que significa que crecerán igualmente para llenar el espacio disponible en el contenedor. Esto crea un diseño flexible y responsivo que se adapta al tamaño del contenedor.

**Resumen**

Flexbox es una herramienta poderosa para crear layouts flexibles y responsivos. Con propiedades como flex-direction, justify-content, y align-items, puedes controlar la disposición y alineación de los elementos de manera eficiente. Flexbox simplifica la creación de diseños complejos y adaptativos, mejorando la experiencia del usuario en diferentes dispositivos y tamaños de pantalla.