**Guía Completa de Flexbox en CSS**

**Introducción a Flexbox**

**Flexbox** es un modelo de diseño en CSS que facilita la alineación, distribución de espacio y el control de la disposición de los elementos dentro de un contenedor flexible. A diferencia de otros modelos como el modelo de cajas tradicional o el sistema de cuadrícula (Grid), Flexbox fue diseñado para poder alinear los elementos en una sola dimensión, ya sea **de forma horizontal** (en una fila) o **vertical** (en una columna).

**¿Por qué usar Flexbox?**

Flexbox es especialmente útil cuando necesitas:

* Distribuir espacio de manera uniforme entre elementos.
* Alinear elementos en el centro de un contenedor.
* Controlar la alineación de los elementos tanto en la dirección de la fila como en la columna.
* Adaptar el diseño para diferentes tamaños de pantalla (responsive design).

**Conceptos Básicos de Flexbox**

**1. Contenedor Flex (Flex Container)**

Un **contenedor flex** es el elemento principal donde se aplica Flexbox. Para activar Flexbox, se debe usar la propiedad display: flex; en el contenedor. Esto convierte a los elementos hijos en **elementos flexibles** (o **flex items**).

css

Copiar código

.contenedor {

display: flex;

}

**2. Elementos Flex (Flex Items)**

Los **elementos flex** son los hijos del contenedor flex y son los que se organizarán según las reglas de Flexbox. Estos elementos se colocan en fila o columna y se pueden alinear y distribuir dentro del contenedor.

**Propiedades del Contenedor Flex**

**1. display**

Esta propiedad activa el modo Flexbox en el contenedor. Los valores más comunes son:

* **display: flex;**: Coloca los elementos en una fila (por defecto).
* **display: inline-flex;**: Similar a flex, pero el contenedor se comporta como un elemento en línea, es decir, no ocupa toda la anchura de la página.

**2. flex-direction**

Define la dirección de los elementos flex dentro del contenedor. Puede tomar los siguientes valores:

* **row (por defecto)**: Los elementos se organizan en una fila (horizontalmente).
* **row-reverse**: Los elementos se organizan en una fila, pero en el orden inverso.
* **column**: Los elementos se organizan en una columna (verticalmente).
* **column-reverse**: Los elementos se organizan en una columna, pero en el orden inverso.

Ejemplo:

css

Copiar código

.contenedor {

display: flex;

flex-direction: row; /\* Cambia a column para ver los elementos en columna \*/

}

**3. flex-wrap**

Controla si los elementos deben envolver en una nueva línea cuando el espacio es insuficiente. Los valores son:

* **nowrap (por defecto)**: Los elementos no se envuelven y permanecen en una sola línea.
* **wrap**: Los elementos se envuelven en una nueva línea cuando no hay espacio suficiente.
* **wrap-reverse**: Los elementos se envuelven, pero el orden de las líneas se invierte.

Ejemplo:

css

Copiar código

.contenedor {

display: flex;

flex-wrap: wrap;

}

**4. justify-content**

Controla la distribución horizontal de los elementos dentro del contenedor. Los valores posibles son:

* **flex-start**: Alinea los elementos al principio del contenedor (por defecto).
* **flex-end**: Alinea los elementos al final del contenedor.
* **center**: Centra los elementos dentro del contenedor.
* **space-between**: Distribuye los elementos dejando el mayor espacio posible entre ellos.
* **space-around**: Distribuye los elementos dejando espacio entre ellos, pero también en los bordes.
* **space-evenly**: Distribuye los elementos dejando un espacio igual entre todos los elementos, incluidos los bordes.

Ejemplo:

css

Copiar código

.contenedor {

display: flex;

justify-content: center; /\* Puedes probar con los otros valores \*/

}

**5. align-items**

Controla la alineación vertical (en el eje cruzado) de los elementos dentro del contenedor. Los valores son:

* **flex-start**: Alinea los elementos al principio del contenedor.
* **flex-end**: Alinea los elementos al final del contenedor.
* **center**: Centra los elementos verticalmente.
* **baseline**: Alinea los elementos según su línea de base (generalmente útil para textos).
* **stretch (por defecto)**: Estira los elementos para que llenen todo el contenedor.

Ejemplo:

css

Copiar código

.contenedor {

display: flex;

align-items: center; /\* Cambia a flex-start o flex-end para probar \*/

}

**6. align-content**

Controla el espacio entre las líneas de los elementos cuando hay múltiples líneas de elementos (si se utiliza flex-wrap). Los valores posibles son:

* **flex-start**: Alinea las líneas al principio del contenedor.
* **flex-end**: Alinea las líneas al final del contenedor.
* **center**: Centra las líneas dentro del contenedor.
* **space-between**: Distribuye las líneas dejando el mayor espacio posible entre ellas.
* **space-around**: Distribuye las líneas dejando espacio entre ellas, pero también en los bordes.
* **stretch (por defecto)**: Estira las líneas para que llenen todo el contenedor.

**Propiedades de los Elementos Flex**

Estas propiedades se aplican a los elementos dentro del contenedor flex y permiten controlar su tamaño, alineación y comportamiento dentro del contenedor.

**1. flex-grow**

Determina cuánto debe crecer un elemento en relación con los demás elementos dentro del contenedor, si hay espacio disponible. El valor predeterminado es 0, lo que significa que los elementos no crecen.

Ejemplo:

css

Copiar código

.item {

flex-grow: 1; /\* Este elemento ocupará todo el espacio disponible \*/

}

**2. flex-shrink**

Controla cuánto puede reducirse un elemento en relación con los demás elementos dentro del contenedor, si el espacio es insuficiente. El valor predeterminado es 1, lo que significa que los elementos se reducirán si es necesario.

Ejemplo:

css

Copiar código

.item {

flex-shrink: 0; /\* Este elemento no se reducirá \*/

}

**3. flex-basis**

Especifica el tamaño inicial de un elemento antes de que se aplique flex-grow o flex-shrink. Puedes establecer un valor en píxeles o cualquier otra unidad de medida.

Ejemplo:

css

Copiar código

.item {

flex-basis: 200px; /\* El tamaño inicial del elemento será de 200px \*/

}

**4. flex**

Es una propiedad abreviada que combina flex-grow, flex-shrink y flex-basis. La sintaxis es:

css

Copiar código

flex: <flex-grow> <flex-shrink> <flex-basis>;

Por ejemplo:

css

Copiar código

.item {

flex: 1 1 200px; /\* Crecerá, se reducirá y su tamaño base será 200px \*/

}

**5. align-self**

Permite que un elemento individual se alinee de manera diferente a los demás elementos en el contenedor, sin afectar a los demás. Los valores son los mismos que para align-items.

Ejemplo:

css

Copiar código

.item {

align-self: center; /\* Este elemento se alineará en el centro \*/

}

**Ejercicios Prácticos**

**Ejercicio 1: Centrar Elementos en el Contenedor**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

**Objetivo:**

Crear un contenedor con tres elementos, centrados tanto horizontal como verticalmente.

**Código HTML y CSS**

**HTML:**

html

Copiar código

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Ejercicio 1</title>

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

</head>

<body>

<div class="contenedor">

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

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

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

</div>

</body>

</html>

**CSS:**

css

Copiar código

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body, html {

height: 100%;

display: flex;

justify-content: center;

align-items: center;

}

.contenedor {

display: flex;

justify-content: center;

align-items: center;

width: 300px;

height: 300px;

background-color: #f0f0f0;

}

.item {

background-color: #4CAF50;

color: white;

padding: 10px;

margin: 10px;

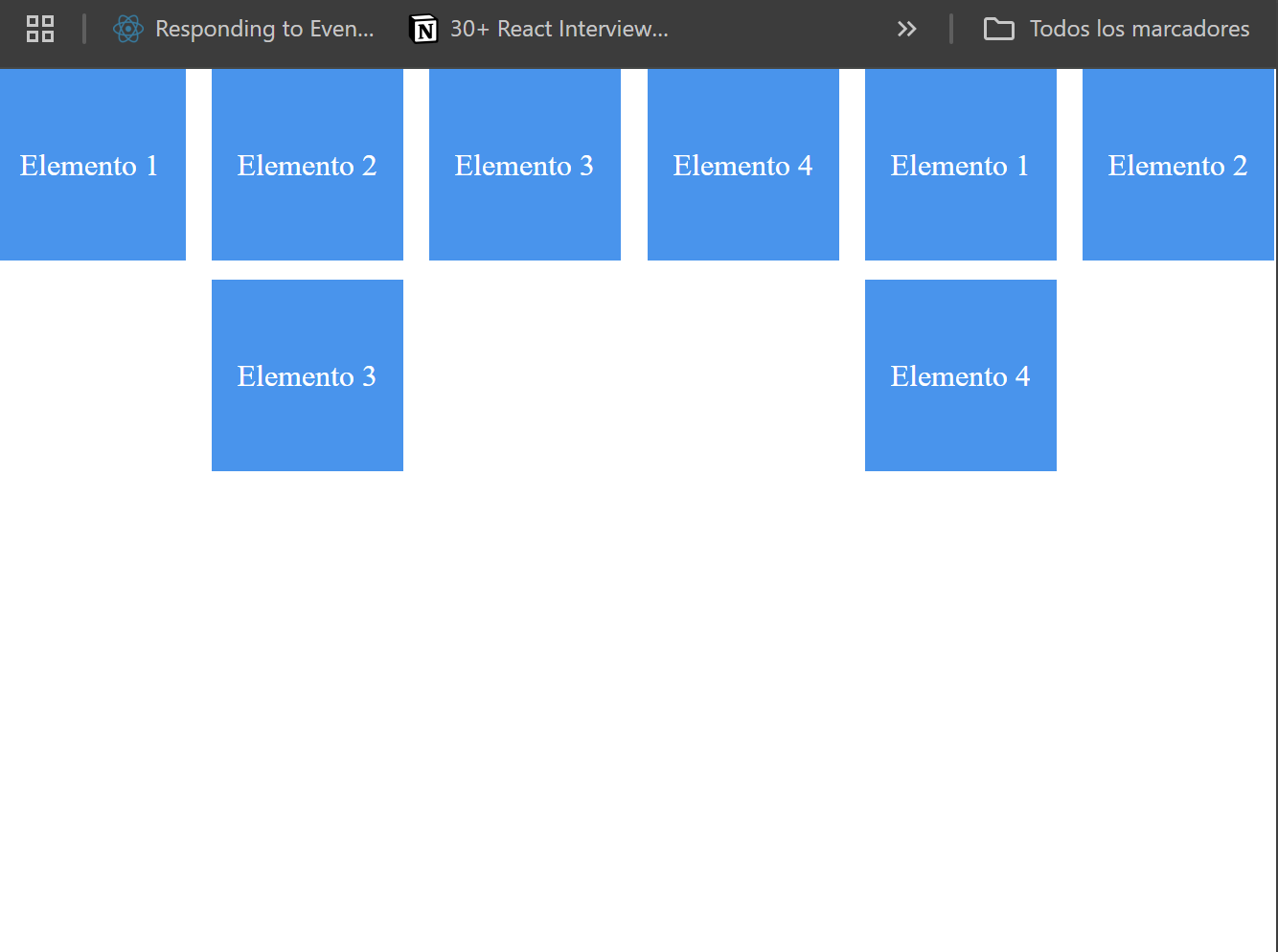
text-align: center;

}

**Ejercicio 2: Disposición de Elementos en Filas y Columnas**

**Objetivo:**

Crear un contenedor con 8 elementos. Disponerlos en filas cuando haya espacio suficiente, y hacer que se envuelvan a columnas cuando el tamaño del contenedor sea más pequeño.



**Código HTML y CSS**

**HTML:**

html

Copiar código

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Ejercicio 2</title>

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

</head>

<body>

<div class="contenedor">

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

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

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

<div class="item">Elemento 4</div>

</div>

</body>

</html>

**CSS:**

css

Copiar código

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body, html {

height: 100%;

}

.contenedor {

display: flex;

flex-wrap: wrap;

justify-content: space-around;

gap: 10px;

}

.item {

width: 100px;

height: 100px;

background-color: #2196F3;

color: white;

text-align: center;

line-height: 100px;

}

**Ejercicio 3: Distribución Uniforme de Elementos**

**Objetivo:**

Distribuir 5 elementos en un contenedor con espacio uniforme entre ellos.

Gráfico, Gráfico en cascada

Descripción generada automáticamente

**Código HTML y CSS**

**HTML:**

html

Copiar código

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Ejercicio 3</title>

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

</head>

<body>

<div class="contenedor">

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

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

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

<div class="item">Elemento 4</div>

<div class="item">Elemento 5</div>

</div>

</body>

</html>

**CSS:**

css

Copiar código

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body, html {

height: 100%;

}

.contenedor {

display: flex;

justify-content: space-between;

width: 500px;

}

.item {

width: 60px;

height: 60px;

background-color: #FF5722;

color: white;

text-align: center;

line-height: 60px;

}

**Ejercicios Propuestos para Practicar**

1. **Ejercicio 4:** Crea una galería de imágenes con Flexbox. Asegúrate de que las imágenes se adapten al tamaño del contenedor y se alineen de manera uniforme.

Interfaz de usuario gráfica, Tabla, Teams

Descripción generada automáticamente

1. **Ejercicio 5:** Realiza un diseño de barra de navegación en horizontal, con los enlaces distribuidos uniformemente. Asegúrate de que el texto de los enlaces esté centrado en su contenedor.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Soluciones

1. Html
2. <!DOCTYPE html>
3. <html lang="en">
4. <head>
5. <meta charset="UTF-8" />
6. <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7. <title>Ejercicio 4: Galería de Imágenes</title>
8. <link rel="stylesheet" href="styles.css" />
9. </head>
10. <body>
11. <div class="galeria">
12. <img src="https://placehold.co/300x300" alt="Imagen 1" />
13. <img src="https://placehold.co/300x300" alt="Imagen 1" />
14. <img src="https://placehold.co/300x300" alt="Imagen 1" />
15. <img src="https://placehold.co/300x300" alt="Imagen 1" />
16. <img src="https://placehold.co/300x300" alt="Imagen 1" />
17. <img src="https://placehold.co/300x300" alt="Imagen 1" />
18. <img src="https://placehold.co/300x300" alt="Imagen 1" />
19. <img src="https://placehold.co/300x300" alt="Imagen 1" />
20. </div>
21. </body>
22. </html>

Css

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

}

body, html {

  height: 100%;

}

.galeria {

  display: flex;

  flex-wrap: wrap;

  justify-content: center;

  gap: 10px;

  padding: 20px;

}

img {

  /\* width: calc(33% - 10px); Tres imágenes por fila, ajustando con margen \*/

  width: 20%;

  height: auto;

  object-fit: cover;

  border-radius: 8px;

}

Ejercicios 2

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>Ejercicio 4: Galería de Imágenes</title>

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

  </head>

  <body>

    <div class="galeria">

      <img src="https://placehold.co/300x300" alt="Imagen 1" />

      <img src="https://placehold.co/300x300" alt="Imagen 1" />

      <img src="https://placehold.co/300x300" alt="Imagen 1" />

      <img src="https://placehold.co/300x300" alt="Imagen 1" />

      <img src="https://placehold.co/300x300" alt="Imagen 1" />

      <img src="https://placehold.co/300x300" alt="Imagen 1" />

      <img src="https://placehold.co/300x300" alt="Imagen 1" />

      <img src="https://placehold.co/300x300" alt="Imagen 1" />

    </div>

  </body>

</html>

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

}

body, html {

  height: 100%;

}

.galeria {

  display: flex;

  flex-wrap: wrap;

  justify-content: center;

  gap: 10px;

  padding: 20px;

}

img {

  /\* width: calc(33% - 10px); Tres imágenes por fila, ajustando con margen \*/

  width: 20%;

  height: auto;

  object-fit: cover;

  border-radius: 8px;

}