**Introducción a Flexbox en 25 Pasos en el Aula 12.**

Flexbox es una técnica moderna de maquetación en CSS que permite distribuir y alinear elementos de forma flexible y eficiente, especialmente útil para crear diseños responsivos.

**1. Flex Container y Flex Items**

* **Flex Container**: El contenedor que agrupa los elementos flexibles. En el video, se crea un div llamado flex-container.
* **Flex Items**: Los elementos dentro del contenedor, que en el ejemplo son varios div numerados.

**2. Propiedades del Flex Container**

* **display: flex**: Activa el modo Flexbox para el contenedor.
* **flex-direction**: Define la dirección de los elementos dentro del contenedor:
  + **row (predeterminado)**: Los elementos se disponen en fila.
  + **column**: Los elementos se disponen en columna.
  + **row-reverse y column-reverse**: Invierten el orden de los elementos en fila o columna.
* **flex-wrap**: Controla si los elementos se ajustan o saltan a una nueva fila cuando no caben:
  + **nowrap**: Los elementos se mantienen en una sola línea (predeterminado).
  + **wrap**: Los elementos saltan a una nueva fila cuando no caben.
  + **wrap-reverse**: Los elementos se envuelven, pero comenzando desde abajo.

**3. Propiedades de Alineación**

* **justify-content**: Alinea los elementos horizontalmente dentro del contenedor. Los valores más comunes incluyen:
  + **flex-start**: Alinea los elementos al inicio del contenedor.
  + **center**: Centra los elementos en el contenedor.
  + **flex-end**: Alinea los elementos al final del contenedor.
  + **space-between**: Distribuye los elementos con espacio igual entre ellos.
  + **space-around**: Distribuye los elementos con espacio igual alrededor de ellos.
* **align-items**: Alinea los elementos verticalmente. Los valores son:
  + **flex-start**: Alinea los elementos en la parte superior del contenedor.
  + **center**: Centra los elementos verticalmente.
  + **flex-end**: Alinea los elementos en la parte inferior.
  + **stretch**: Estira los elementos para llenar el contenedor.
  + **baseline**: Alinea los elementos según su línea base de texto.
* **align-content**: Controla el espacio entre filas múltiples. Valores como:
  + **space-between**: Deja espacio entre las filas.
  + **space-around**: Deja espacio alrededor de las filas.
  + **center**: Centra las filas en el contenedor verticalmente.

**4. Propiedades de los Flex Items**

* **order**: Cambia el orden visual de los elementos sin modificar su orden en el código HTML.
* **flex-grow**: Controla cuánto puede crecer un elemento en proporción a los otros elementos.
* **flex-shrink**: Controla cuánto puede reducirse un elemento cuando es necesario ajustar el espacio.
* **flex-basis**: Define el tamaño inicial de un elemento antes de aplicar flex-grow o flex-shrink.
* **flex**: Es una propiedad abreviada que combina flex-grow, flex-shrink y flex-basis en una sola línea.

**5. Propiedad combinada flex-flow**

* Combina las propiedades flex-direction y flex-wrap. Por ejemplo:
  + **flex-flow: row wrap;**: Los elementos se disponen en fila y envuelven cuando no caben.
  + **flex-flow: column wrap;**: Los elementos se disponen en columna y envuelven.

**6. Ejemplos de Uso Práctico**

En el video, se muestra cómo se aplican estas propiedades paso a paso. Algunos de los pasos incluyen:

* Aplicar flex-direction: row-reverse; para invertir la disposición de los elementos.
* Utilizar flex-wrap: wrap-reverse; para que los elementos envuelvan desde abajo hacia arriba.
* Usar justify-content: space-around; y space-between para distribuir los elementos con diferentes espacios entre ellos.
* Probar varias combinaciones de align-items, como flex-start, flex-end, stretch, y baseline.

**1º Paso**

\*{

box-sizing: border-box;

}

.flex-container{

display: flex;

background-color: brown;

flex-direction: column;

padding: 10px;

border: 2px solid gray;

}

.flex-container div{

background-color: white;

width: 100px;

//width: 100%;//

margin: 10px;

text-align: center;

line-height: 75px;

font-size: 30px;

border: 2px solid black;

}

* display: flex;: Establece un contenedor flexible.
* flex-direction: column;: Los elementos se organizan en una columna (uno debajo del otro).
* Este es un buen punto de partida si quieres que los elementos se coloquen verticalmente.

**Alternativa**:

* Podrías usar row si prefieres que los elementos se coloquen en una fila horizontal, que es el valor predeterminado.

**2º Paso**

flex-direction: row-reverse;

**Explicación**:

* Cambia la dirección de los elementos, colocándolos en una fila pero en orden inverso (de derecha a izquierda).

**Alternativa**:

* Podrías usar row para mantenerlos en el orden normal (de izquierda a derecha).

**3º Paso**

Lo quitamos

flex-direction: row-reverse;

* Simplemente vuelves al comportamiento predeterminado, donde los elementos se alinean en fila de izquierda a derecha.

**4º Paso**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="style2.css">

</head>

<body>

<div class="flex-container">

<div>1</div>

<div>2</div>

<div>3</div>

<div>4</div>

<div>5</div>

<div>6</div>

<div>7</div>

<div>8</div>

<div>9</div>

<div>10</div>

<div>11</div>

<div>12</div>

</div>

</body>

</html>

**Explicación**:

* flex-wrap: wrap;: Permite que los elementos se envuelvan si no caben en una sola línea.
* Útil cuando tienes muchos elementos y quieres que pasen a una nueva fila si la pantalla es pequeña.

**Alternativa**:

* flex-wrap: nowrap;: Fuerza que todos los elementos estén en una sola línea, aunque se desborden.

**5º Paso**

\*{

box-sizing: border-box;

}

.flex-container{

display: flex;

background-color: brown;

flex-wrap: wrap;

padding: 10px;

border: 2px solid gray;

}

.flex-container div{

background-color: white;

width: 100px;

/\* width: 100%; \*/

margin: 10px;

text-align: center;

line-height: 75px;

font-size: 30px;

border: 2px solid black;

}

**Explicación**:

* flex-wrap: wrap;: Permite que los elementos se envuelvan si no caben en una sola línea.
* Útil cuando tienes muchos elementos y quieres que pasen a una nueva fila si la pantalla es pequeña.

**Alternativa**:

* flex-wrap: nowrap;: Fuerza que todos los elementos estén en una sola línea, aunque se desborden.

**6º Paso:**

Flex-wrap: wrap-reverse;

**Explicación**:

* Al igual que wrap, pero los elementos que se envuelven aparecerán en la parte superior en lugar de en la parte inferior.

**7º Paso:**

\*{

box-sizing: border-box;

}

.flex-container{

display: flex;

background-color: brown;

flex-flow: row wrap;

padding: 10px;

border: 2px solid gray;

}

.flex-container div{

background-color: white;

width: 100px;

/\* width: 100%; \*/

margin: 10px;

text-align: center;

line-height: 75px;

font-size: 30px;

border: 2px solid black;

}

**Explicación**:

* Combina flex-direction: row; y flex-wrap: wrap; en una sola propiedad (flex-flow).
* Organiza los elementos en una fila y permite que se envuelvan.

**8º Paso**

flex-flow: column wrap;

**Explicación**:

* Organiza los elementos en una columna y permite que se envuelvan verticalmente.

**9º Paso**

flex-flow: row wrap;

**Explicación**:

* Regresa a la disposición de fila con capacidad de envolver, como en el 6º paso

**10º Paso**

\*{

box-sizing: border-box;

}

.flex-container{

display: flex;

background-color: brown;

justify-content: center;

padding: 10px;

border: 2px solid gray;

}

.flex-container div{

background-color: white;

width: 100px;

/\* width: 100%; \*/

margin: 10px;

text-align: center;

line-height: 75px;

font-size: 30px;

border: 2px solid black;

}

**Explicación**:

* Centra los elementos horizontalmente dentro del contenedor flex.

**Alternativa**:

* justify-content: flex-start;: Alinea los elementos al principio.
* justify-content: flex-end;: Alinea los elementos al final.

**11º Paso**

justify-content: space-around;

justify-content: space-between;

**12ºpaso**

\*{

box-sizing: border-box;

}

.flex-container{

display: flex;

height: 400px;

background-color: brown;

padding: 10px;

border: 2px solid gray;

}

.flex-container div{

background-color: white;

width: 100px;

/\* width: 100%; \*/

margin: 10px;

text-align: center;

line-height: 75px;

font-size: 30px;

border: 2px solid black;

}

**Explicación**:

* Controla cómo los elementos se alinean verticalmente dentro del contenedor:
  + flex-start: Los alinea al principio del eje vertical.
  + flex-end: Los alinea al final del eje vertical.
  + stretch: Estira los elementos para que ocupen todo el espacio vertical disponible.
  + baseline: Los alinea a lo largo de la línea base del texto.

**13ºPaso**

align-items: flex-start;

align-items: flex-end;

align-items: Stretch;

align-items: baseline;

**Explicación**:

* Establece un tamaño de fuente de 80px para el primer elemento div. Puedes controlar fácilmente el tamaño individual de cada div usando estilos en línea.

**14ºPaso**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="style2.css">

</head>

<body>

<div class="flex-container">

<div style="font-size: 80px;">1</div>

<div>2</div>

<div>3</div>

<div>4</div>

</div>

</body>

</html>

**Explicación**:

* Cada div tiene un tamaño de fuente diferente definido en línea, lo que le da a cada elemento un tamaño único.

**15º Paso**

<div style="font-size: 80px;">1</div>

<div style="font-size: 20px;">2</div>

<div style="font-size: 15px;">3</div>

<div style="font-size: 60px;">4</div>

**16º Paso**

<div class="flex-container">

<div>1</div>

<div>2</div>

<div>3</div>

<div>4</div>

<div>5</div>

<div>6</div>

<div>7</div>

<div>8</div>

<div>9</div>

<div>10</div>

<div>11</div>

<div>12</div>

</div>

.flex-container {

display: flex;

height: 400px;

flex-wrap: wrap;

align-content: space-between;

background-color: brown;

padding: 10px;

border: 2px solid gray;

}

.flex-container div {

background-color: white;

width: 100px;

/\* width: 100%; \*/

margin: 10px;

text-align: center;

line-height: 75px;

font-size: 30px;

border: 2px solid black;

}

**Explicación**:

* align-content: Controla la distribución de las filas en el eje transversal (vertical en un layout de row).
  + space-around: Coloca espacio alrededor de las filas.
  + space-between: Coloca espacio entre las filas.

**17º Paso**

align-content : space-around;

align-content : space-between;

**Explicación**:

* stretch: Hace que las filas se estiren para ocupar todo el espacio vertical disponible.
* center: Centra las filas verticalmente en el contenedor.

**18º Paso**

<body>

    <div class="flex-container">

        <div>1</div>

        <div>2</div>

        <div>3</div>

        <div>4</div>

        <div>5</div>

        <div>6</div>

        <div>7</div>

        <div>8</div>

        <div>9</div>

        <div>10</div>

    </div>

**19º Paso**

 align-content : stretch;

Intenta ocupar todo el alto que puede

 align-content : center;

Centra todo el contenido

 align-content : flex-start;

 align-content : flex-end;

**Explicación**:

* Aplicas colores de fondo específicos para cada div, lo que mejora la visibilidad de cada caja dentro del contenedor flex.

**20º Paso**

Elementos hijos:

<div class="flex-container">

<div style="background-color: red;">1</div>

<div style="background-color: blue;">2</div>

<div style="background-color: green;">3</div>

<div style="background-color: yellow;">4</div>

</div>

**Explicación**:

* Usas la propiedad order para cambiar el orden en que los elementos se muestran, independientemente de su posición en el HTML.

**Alternativa**:

* Los elementos con el mismo order se alinearán según el orden de aparición en el HTML.

**21º Paso**

  <div class="flex-container">

            <div style="order: 1; background-color: red;">1</div>

             <div style="order: 2;background-color: blue;">2</div>

             <div style="order: 3;background-color: green;">3</div>

            <div style="order: 4;background-color: yellow;">4</div>

            </div>

Le damos un orden según otorguemos.

**22º Paso**

<div class="flex-container">

            <div style="flex-grow: 1; background-color: red;">1</div>

             <div style="flex-grow: 1;background-color: blue;">2</div>

             <div style="flex-grow: 1;background-color: green;">3</div>

            <div style="flex-grow: 1;background-color: yellow;">4</div>

            </div>

**Explicación**:

* flex-grow: 1;: Permite que el elemento crezca para ocupar el espacio restante dentro del contenedor. Si todos los elementos tienen flex-grow: 1, ocuparán el mismo espacio disponible.

**Alternativa**:

* Puedes ajustar el valor de flex-grow para hacer que algunos elementos crezcan más que otros (por ejemplo, flex-grow: 2).

**23º Paso**

        <div class="flex-container">

            <div style="flex-shrink: 10; background-color: red;">1</div>

             <div style="flex-grow: 2;background-color: blue;">2</div>

             <div style="flex-grow: 4;background-color: green;">3</div>

            <div style="flex-grow: 6;background-color: yellow;">5</div>

            <div style="flex-grow: 8;background-color: blue;">6</div>

             <div style="flex-grow: 10;background-color: green;">7</div>

            <div style="flex-grow: 12;background-color: yellow;">8</div>

            </div>

**24º Paso**

<div class="flex-container">

            <div style="flex-basis: 10px; background-color: red;">1</div>

             <div style="flex-grow: 2;background-color: blue;">2</div>

             <div style="flex-grow: 4;background-color: green;">3</div>

            <div style="flex-grow: 6;background-color: yellow;">5</div>

            <div style="flex-grow: 8;background-color: blue;">6</div>

             <div style="flex-grow: 10;background-color: green;">7</div>

            <div style="flex-grow: 12;background-color: yellow;">8</div>

            </div>

**25º Paso**

 <div class="flex-container">

            <div style="flex: 0 0 100px; background-color: red;">1</div>

             <div style="flex-grow: 2;background-color: blue;">2</div>

             <div style="flex-grow: 4;background-color: green;">3</div>

            <div style="flex-grow: 6;background-color: yellow;">5</div>

            <div style="flex-grow: 8;background-color: blue;">6</div>

             <div style="flex-grow: 10;background-color: green;">7</div>

            <div style="flex-grow: 12;background-color: yellow;">8</div>

            </div>