



Flexbox, modelo de caja flexible en CSS

-
- C.F.G.S. DAW
 - 6 horas semanales
 - Mes aprox. de impartición: Nov - Dic
 - iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- **Analizar y seleccionar los colores y las tipografías adecuados para la visualización en la pantalla.**
- **Utilizar marcos y tablas para presentar la información de manera ordenada.**
- **Identificar nuevos elementos y etiquetas en HTML5.**
- **Reconocer las posibilidades de modificar etiquetas HTML.**
- **Valorar la utilidad de las hojas de estilo para conseguir un diseño uniforme en todo el sitio web.**

GLOSARIO



Formularios. Documentos interactivos utilizados para recoger información en un sitio web. Esta información es enviada al servidor, donde es procesada. Cada formulario contiene uno o varios tipos de controles que permiten recolectar la información de varias formas diferentes.

Fuentes seguras. Fuentes tipográficas que los usuarios tenían instaladas por defecto en su dispositivo. En la actualidad, gracias a que la mayoría de los navegadores soportan la directiva @font-face, es posible utilizar casi cualquier tipografía a través de Google Fonts.

Guías de estilo. Documentos con directrices que permiten la normalización de estilos. En estas guías se recogen los criterios y normas que debe seguir un proyecto; de esta forma se ofrece una apariencia más uniforme y atractiva para el usuario.

HTML. Lenguaje de marcado de hipertexto utilizado en las páginas web. Este tipo de lenguaje presenta una forma estructurada y agradable, con hipervínculos que conducen a otros documentos y con inserciones multimedia (sonido, imágenes, vídeos...).

GLOSARIO



HTML5. Última versión del lenguaje para la programación de páginas web HTML. Los sitios implementados con este lenguaje solo pueden visualizarse correctamente en los navegadores más actuales.

Legibilidad. Cualidad deseable en una familia tipográfica. Se trata de la facilidad de la lectura de una letra. Esta cualidad puede venir determinada por varios parámetros como el interletrado, el interpalabrado o el interlineado.

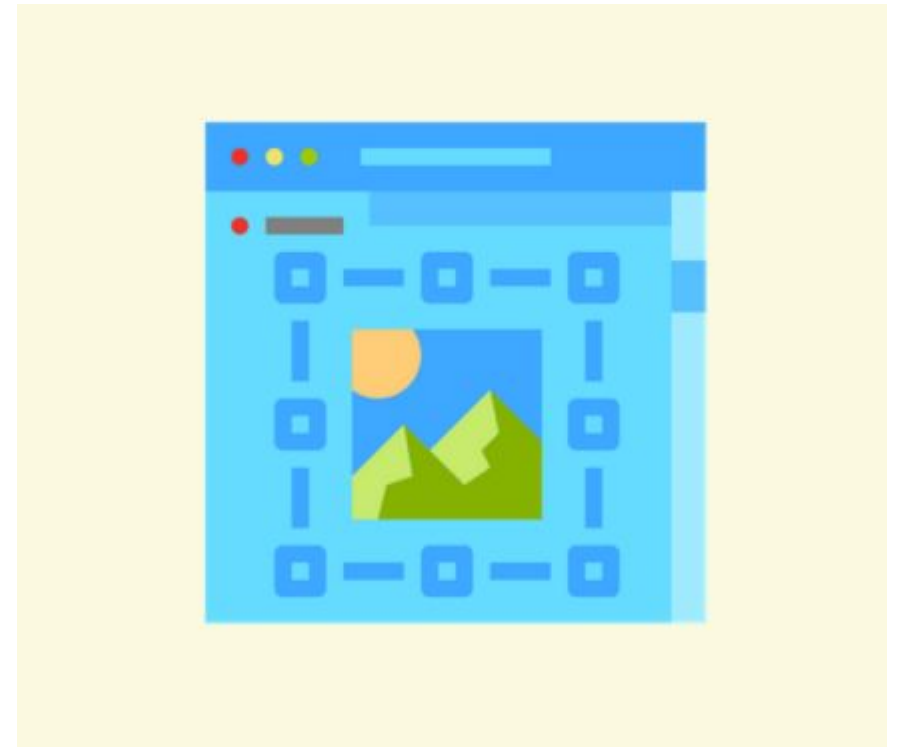
Marcos. Son las ventanas independientes incorporadas dentro de la página general. Gracias a ellos, cada página quedará dividida en varias subpáginas, permitiendo realizar un diseño más organizado y limpio a la vista. Con HTML5 ha quedado obsoleto.

Tipografía. Se trata del tipo de letra que se escoge para un determinado diseño. Según la RAE, significa "modo o estilo en que está impreso un texto" o "clase de tipos de imprenta".

INTRODUCCIÓN



Flexbox, también conocido como **modelo de caja flexible**, es un diseño unidimensional que permite crear un diseño más eficiente y predecible en situaciones donde el tamaño de los elementos es desconocido o dinámico. Este modelo flexible no necesita utilizar los métodos tradicionales de posicionamiento de cajas (static, relative, absolute o float) sino que organiza los elementos utilizando **contenedores flexibles**.



PROPIEDADES *Flexbox* CSS



Diseño unidimensional: Flexbox trabaja en una dimensión a la vez, ya sea en fila o en columna. Esto permite una fácil alineación y distribución de elementos a lo largo de un eje principal (main axis) y un eje transversal (cross axis).

Para utilizar Flexbox, primero debes declarar un contenedor como un contenedor flexible utilizando `display: flex` o `display: inline-flex`. Los elementos hijos de este contenedor se convierten automáticamente en elementos flexibles.

PROPIEDADES *Flexbox* CSS

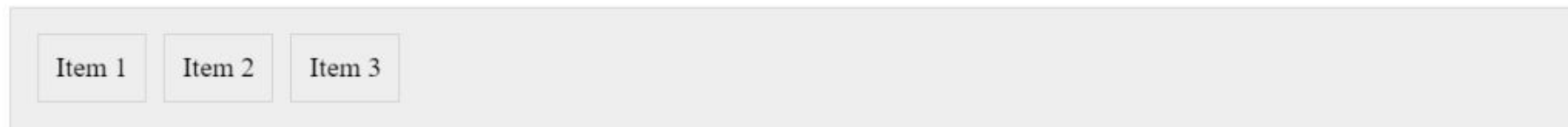
display: flex | inline-flex



La propiedad **display** en CSS determina cómo se muestra un elemento en el diseño de la página. En el contexto de flexbox, estos dos valores, **flex** e **inline-flex**, se utilizan para establecer un contenedor como un contenedor flexible.

DIFERENCIA VISUAL ENTRE LOS VALORES *FLEX* E *INLINE-FLEX*:

display: flex



display: inline-flex



PROPIEDADES *Flexbox* CSS

display: flex | inline-flex



- **flex:** Cuando aplicas `display: flex` a un contenedor, este se convierte en un contenedor flexible que distribuye su contenido de manera flexible a lo largo del eje principal. Por defecto, el eje principal es horizontal (de izquierda a derecha). Los elementos secundarios dentro de este contenedor flex se comportarán como bloques y se ajustarán automáticamente para llenar el contenedor en la dirección principal.

```
.contenedor-flex {  
  display: flex;  
}
```

PROPIEDADES *Flexbox* CSS

display: flex | inline-flex



- **inline-flex**: Este valor es similar a `display: flex`, pero el contenedor flexible se comporta como un elemento en línea en lugar de un bloque. Esto significa que ocupará solo el ancho necesario y permitirá que otros elementos en línea coexistan en la misma línea.

```
.contenedor-inline-flex {  
  display: inline-flex;  
}
```

PROPIEDADES *Flexbox* CSS

display: flex | inline-flex



Código HTML:

```
<h2>display: flex</h2>
<div class="contenedor flex-container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

```
<h2>display: inline-flex</h2>
<div class="contenedor
inline-flex-container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

PROPIEDADES *Flexbox* CSS

display: flex | inline-flex



Código CSS:

```
.contenedor {  
    border: 1px solid #ccc;  
    background-color: #EEE;  
    padding: 10px;  
}
```

```
.flex-container {  
    display: flex;  
}
```

```
.inline-flex-container {  
    display: inline-flex;  
}
```

```
.item {  
    border: 1px solid #ccc;  
    padding: 10px;  
    margin: 5px;  
}
```

PROPIEDADES *Flexbox* CSS


display: flex | inline-flex



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:


<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/MYqaLog>

display: flex



Item 1 Item 2 Item 3

display: inline-flex



Item 1 Item 2 Item 3

PROPIEDADES *Flexbox* CSS

flex-wrap: nowrap | wrap | wrap-reverse



Podemos indicar qué elementos flexibles se deben trasladar cuando no hay suficiente espacio en el contenedor mediante la propiedad **flex-wrap**. Sus valores son los siguientes:

- **nowrap**: los elementos no pasan a la siguiente fila y se reduce su anchura para mostrarlos. Este es el valor por defecto.
- **wrap**: los elementos pasan a la siguiente fila y conservan su anchura.
- **wrap-reverse**: los elementos pasan a la siguiente fila, pero en sentido inverso al de su declaración.

```
#container{  
    display: flex;  
    flex-wrap: wrap;  
}
```

PROPIEDADES *Flexbox* CSS

flex-wrap: nowrap | wrap | wrap-reverse



Código HTML:

```
<h2>flex-wrap: no-wrap;</h2>
<div id="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
</div>
```

```
<h2>flex-wrap: wrap;</h2>
<div id="flex-container1">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
</div>
<h2>flex-wrap: wrap-reverse;</h2>
<div id="flex-container2">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
</div>
```

PROPIEDADES *Flexbox* CSS

flex-wrap: nowrap | wrap | wrap-reverse



Código CSS:

```
#flex-container{
  display: flex;
  flex-wrap: nowrap;
}
#flex-container div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #E67E22;
}
```

```
#flex-container1{
  display: flex;
  flex-wrap: wrap;
}
#flex-container1 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #C0392B;
}
```


PROPIEDADES *Flexbox* CSS

flex-wrap: nowrap | wrap | wrap-reverse

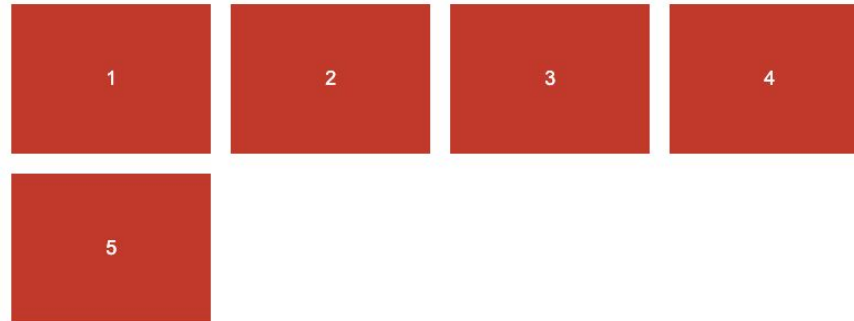


```
#flex-container2{  
  display: flex;  
  flex-wrap: wrap-reverse;  
}  
  
#flex-container2 div{  
  width: 200px;  
  height: 150px;  
  color: #FFF;  
  text-align: center;  
  line-height: 150px;  
  margin: 10px;  
  font-size: 20px;  
  font-family: Arial;  
  background-color: #3498DB;  
}
```

flex-wrap: no-wrap;



flex-wrap: wrap;



flex-wrap: wrap-reverse;



La salida del código anterior
puede verse en la imagen
adjunta o en el siguiente link:

[https://codepen.io/Carmelo-Jos
-Ja-n-D-az/pen/NPKGowo](https://codepen.io/Carmelo-Jos-n-D-az/pen/NPKGowo)

PROPIEDADES *Flexbox* CSS

justify-content: flex-start | flex-end | etc



La propiedad `justify-content` es la que define la justificación horizontal de los elementos hijos de un contenedor flexible. Esta propiedad acepta cinco valores:

- `flex-start`: posiciona los elementos a la izquierda del contenedor. Valor por defecto.
- `flex-end`: posiciona los elementos a la derecha del contenedor.
- `center`: centra los elementos en el contenedor
- `space-between`: añade un espacio idéntico entre los elementos. El primer elemento está alineado a la izquierda del contenedor, y el último, a la derecha.
- `space-around`: espacia de forma regular los elementos que no están alineados a la izquierda y a la derecha del contenedor.

```
#container{    display: flex;    justify-content: flex-start; }
```

PROPIEDADES *Flexbox* CSS

justify-content: flex-start | flex-end | etc



Código HTML:

```
<h2>justify-content: flex-start;</h2>
<div id="flex-container1">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
<h2>justify-content: flex-end;</h2>
<div id="flex-container2">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```

```
<h2>justify-content: center;</h2>
<div id="flex-container3">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
<h2>justify-content:
space-between;</h2>
<div id="flex-container4">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```

PROPIEDADES *Flexbox* CSS

justify-content: flex-start | flex-end | etc



Código HTML:

```
<h2>justify-content: space-around;</h2>
<div id="flex-container5">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```

PROPIEDADES *Flexbox* CSS

justify-content: flex-start | flex-end | etc



Código CSS:

```
#flex-container1{
  display: flex;
  justify-content: flex-start;
}
#flex-container1 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: 'Roboto', sans-serif;
  background-color: #9B59B6;
}
```

```
#flex-container2{
  display: flex;
  justify-content: flex-end;
}
#flex-container2 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #C0392B;
}
```

PROPIEDADES *Flexbox* CSS

justify-content: flex-start | flex-end | etc



Código CSS:

```
#flex-container3{
  display: flex;
  justify-content: center;
}
#flex-container3 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #3498DB;
}
```

```
#flex-container4{
  display: flex;
  justify-content: space-between;
}
```

```
#flex-container4 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #17A589;
}
```

```
#flex-container5{
```

PROPIEDADES *Flexbox* CSS

justify-content: flex-start | flex-end | etc



Código CSS:

```
#flex-container5{
  display: flex;
  justify-content: space-around;
}
#flex-container5 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #E67E22;
}
```

justify-content: flex-start;



justify-content: flex-end;



justify-content: center;



justify-content: space-between;



justify-content: space-around;



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/PwYzYJK>

PROPIEDADES *Flexbox* CSS

align-items: stretch | flex-start | etc



La propiedad `align-items` actúa sobre el eje vertical. Para utilizar esta propiedad primero se debe definir la propiedad `flex-direction: row`.

La propiedad `align-items` acepta cinco valores:

- **stretch**: los elementos se amplían verticalmente para ocupar toda la altura disponible en el contenedor. Este es el valor por defecto.
- **flex-start**: los elementos se colocan en la parte superior del contenedor.
- **flex-end**: los elementos se colocan en la parte inferior del contenedor.
- **center**: los elementos se colocan en el centro del contenedor.
- **baseline**: los elementos se alinean sobre la línea de base del texto.

```
#container{ display: flex; flex-direction: row; align-items: stretch; }
```


PROPIEDADES *Flexbox* CSS

align-items: stretch | flex-start | etc



Código HTML:

```
<h2>align-items: stretch;</h2>
<div id="flex-container1">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
<h2>align-items: flex-start;</h2>
<div id="flex-container2">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```

```
<h2>align-items: flex-end;</h2>
<div id="flex-container3">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
<h2>align-items: center;</h2>
<div id="flex-container4">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```

PROPIEDADES *Flexbox* CSS

align-items: stretch | flex-start | etc



Código HTML:

```
<h2>align-items: baseline;</h2>
<div id="flex-container5">
  <div><h3>1</h3></div>
  <div>2</div>
  <div><h4>3</h4></div>
  <div>4</div>
</div>
```

PROPIEDADES *Flexbox* CSS

align-items: stretch | flex-start | etc



Código CSS:

```
#flex-container1{
  display: flex;
  justify-content: center;
  flex-direction: row;
  align-items: stretch;
  height: 250px;
}
#flex-container1 div{
  width: 200px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: 'Roboto', sans-serif;
  background-color: #9B59B6;
}
```

```
#flex-container2{
  display: flex;
  justify-content: center;
  flex-direction: row;
  align-items: flex-start;
  height: 250px;
}
#flex-container2 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #C0392B;
}
```

PROPIEDADES *Flexbox* CSS

align-items: stretch | flex-start | etc



```
#flex-container3{
  display: flex;
  justify-content: center;
  flex-direction: row;
  align-items: flex-end;
  height: 250px;
}
#flex-container3 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #3498DB;
}
```

```
#flex-container4{
  display: flex;
  justify-content: center;
  flex-direction: row;
  align-items: center;
  height: 250px;
}
#flex-container4 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #17A589;
}
```

PROPIEDADES *Flexbox* CSS

align-items: stretch | flex-start | etc



Código CSS:

```
#flex-container5{  
  display: flex;  
  justify-content: center;  
  flex-direction: row;  
  align-items: baseline;  
  height: 250px;  
}  
#flex-container5 div{  
  width: 200px;  
  height: 150px;  
  color: #FFF;  
  text-align: center;  
  line-height: 120px;  
  margin: 10px;  
  font-size: 20px;  
  font-family: Arial;  
  background-color: #E67E22;}
```

align-items: stretch;



align-items: flex-start;



align-items: flex-end;



align-items: center;



align-items: baseline;



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/LEPZPdG>

PROPIEDADES *Flexbox* CSS

flex-direction: row | row-reverse | etc



La propiedad **flex-direction** determina la dirección en la que los elementos flexibles se colocan dentro de un contenedor flex. Puede establecerse para organizar los elementos en una fila (horizontal) o en una columna (vertical).

La propiedad **flex-direction** acepta cuatro valores:

- **row (valor por defecto)**: Coloca los elementos en una fila horizontal. El inicio está a la izquierda y el final a la derecha.
- **row-reverse**: Coloca los elementos en una fila horizontal, pero en orden inverso al valor por defecto. El inicio estará a la derecha y el final a la izquierda.
- **column**: Coloca los elementos en una columna vertical. El inicio está arriba y el final abajo.
- **column-reverse**: Coloca los elementos en una columna vertical, pero en orden inverso al valor por defecto. El inicio estará abajo y el final arriba.

```
#container { display: flex; flex-direction: row; }
```

PROPIEDADES *Flexbox* CSS

flex-direction: row | row-reverse | etc



Código HTML:

```
<h2>flex-direction: row;</h2>
<div id="flex-container1">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

```
<h2>flex-direction: row-reverse;</h2>
<div id="flex-container2">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

```
<h2>flex-direction: column;</h2>
<div id="flex-container3">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

```
<h2>flex-direction:
column-reverse;</h2>
<div id="flex-container4">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
```

PROPIEDADES *Flexbox* CSS

flex-direction: row | row-reverse | etc



Código CSS:

```
#flex-container1{
  display: flex;
  flex-direction: row;
}
#flex-container1 div{
  width: 100px;
  height: 50px;
  color: #FFF;
  text-align: center;
  line-height: 50px;
  margin: 10px;
  font-size: 20px;
  font-family: 'Roboto', sans-serif;
  background-color: #9B59B6;
}
```

```
#flex-container2{
  display: flex;
  flex-direction: row-reverse;
}
#flex-container2 div{
  width: 100px;
  height: 50px;
  color: #FFF;
  text-align: center;
  line-height: 50px;
  margin: 10px;
  font-size: 20px;
  font-family: Arial;
  background-color: #C0392B;
}
```


PROPIEDADES *Flexbox* CSS

flex-direction: row | row-reverse | etc



```
#flex-container3{  
  display: flex;  
  flex-direction: column;  
}
```

```
#flex-container3 div{  
  width: 100px;  
  height: 50px;  
  color: #FFF;  
  text-align: center;  
  line-height: 50px;  
  margin: 10px;  
  font-size: 20px;  
  font-family: Arial;  
  background-color: #3498DB;  
}
```

```
#flex-container4{  
  display: flex;  
  flex-direction: column-reverse;  
}
```

```
#flex-container4 div{  
  width: 100px;  
  height: 50px;  
  color: #FFF;  
  text-align: center;  
  line-height: 50px;  
  margin: 10px;  
  font-size: 20px;  
  font-family: Arial;  
  background-color: #17A589;  
}
```

PROPIEDADES *Flexbox* CSS

flex-direction: row | row-reverse | etc



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/jENrNXX>

`flex-direction: row;`



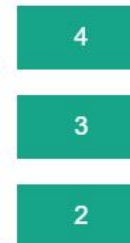
`flex-direction: row-reverse;`



`flex-direction: column;`



`flex-direction: column-reverse;`



OTRAS PROPIEDADES



Las propiedades de *Flexbox* que hemos visto son las más comunes y utilizadas para controlar el comportamiento de los elementos flex. Sin embargo, hay más propiedades disponibles que te permiten tener un mayor control sobre el diseño de tus elementos flex. Algunas de estas propiedades adicionales incluyen `order`, `flex-grow`, `flex-shrink`, y `flex-basis`, entre otras.

Para entender estas propiedades primero necesitamos un contenedor Flexbox:

```
<div class="flex-container">
  <div class="flex-item">Item 1</div>
  <div class="flex-item">Item 2</div>
  <div class="flex-item">Item 3</div>
</div>
```

```
.flex-container {
  display: flex;
}
```

OTRAS PROPIEDADES

flex-grow



`flex-grow` define cómo un elemento debe crecer en relación con el resto de los elementos flexibles en el contenedor Flexbox.

```
.flex-item {  
    flex-grow: 1; /* Todos los elementos crecen de igual manera */  
}
```

Si uno de los elementos tiene `flex-grow: 2`, ese elemento intentará ocupar el doble del espacio que los otros elementos.

OTRAS PROPIEDADES

flex-shrink



`flex-shrink` define cómo los elementos se encogen en relación con los demás cuando no hay suficiente espacio en el contenedor.

```
.flex-container {  
    display: flex;  
    width: 500px;  
    /* Ancho total menor que la suma de los  
    flex-basis */  
}  
  
.flex-item {  
    flex-basis: 200px;  
}
```

```
/* Primer elemento */  
.flex-container .flex-item:first-child  
{  
    flex-shrink: 0;  
    /* Este elemento no se encogerá */  
}  
  
/* Segundo elemento */  
.flex-container .flex-item:nth-child(2)  
{  
    flex-shrink: 2;  
    /* Este elemento se encogerá el doble  
    que el normal */  
}
```

OTRAS PROPIEDADES

flex-shrink



`flex-shrink` define cómo los elementos se encogen en relación con los demás cuando no hay suficiente espacio en el contenedor.

```
.flex-container {
  display: flex;
  width: 500px;
  /* Ancho total menor que la suma de los
  flex-basis */
}

.flex-item {
  flex-basis: 200px;
}
```

```
/* Primer elemento */
.flex-container .flex-item:first-child
{
  flex-shrink: 0;
  /* Este elemento no se encogerá */
}

/* Segundo elemento */
.flex-container .flex-item:nth-child(2)
{
  flex-shrink: 2;
  /* Este elemento se encogerá el doble
  que el normal */
}
```

OTRAS PROPIEDADES

flex-shrink



`flex-shrink` define cómo los elementos se encogen en relación con los demás cuando no hay suficiente espacio en el contenedor.

```
/* Tercer elemento */  
.flex-container .flex-item:nth-child(3) {  
    flex-shrink: 1;  
/* Este elemento se encogerá normalmente */  
}
```

OTRAS PROPIEDADES

flex-basis



`flex-basis` establece el tamaño inicial de un elemento antes de que se distribuya el espacio restante.

```
.flex-item {  
    flex-basis: 100px;  
    /* Cada elemento comienza con 100px de ancho */  
}
```


OTRAS PROPIEDADES

Ejemplos



Código HTML:

```
<h2>Por defecto</h2>
<div class="flex-container">
  <div class="flex-item">Item
1</div>
  <div class="flex-item">Item
2</div>
  <div class="flex-item">Item
3</div>
</div>
```

<h2>flex-grow:1: Todos los
elementos crecen de igual
manera</h2>

```
<div class="flex-container">
  <div class="flex-item
flex-item-1">Item 1</div>
  <div class="flex-item
flex-item-1">Item 2</div>
  <div class="flex-item
flex-item-1">Item 3</div>
</div>
```

OTRAS PROPIEDADES

Ejemplos



Código HTML:

<h2>flex-shrink: Los elementos se encogen según su valor de flex-shrink cuando no caben según el valor de flex-basics</h2>

```
<div class="flex-container2">
  <div class="flex-item
flex-item2">Item 1</div>
  <div class="flex-item
flex-item2">Item 2</div>
  <div class="flex-item
flex-item2">Item 3</div>
</div>
```

<h2>flex-basis: establece el tamaño inicial de un elemento antes de que se distribuya el espacio restante.</h2>

```
<div class="flex-container">
  <div class="flex-item
flex-item3">Item 1</div>
  <div class="flex-item
flex-item3">Item 2</div>
  <div class="flex-item
flex-item3">Item 3</div>
</div>
```

OTRAS PROPIEDADES

Ejemplos



Código CSS:

```
.flex-container {
    display: flex;
}
.flex-item {
    background-color: #EEE;
    padding: 10px;
    margin: 10px;
}
.flex-item-1 {
    flex-grow: 1; /* Todos los
elementos crecen de igual manera */
}
```

```
.flex-container2 {
    display: flex;
    width: 500px; /* Ancho total
menor que la suma de los flex-basis */
}
.flex-item2 {
    flex-basis: 200px;
}
/* Primer elemento */
.flex-container2
.flex-item2:first-child {
    flex-shrink: 0; /* Este elemento
no se encogerá */
}
```

OTRAS PROPIEDADES

Ejemplos



Código CSS:

```
/* Segundo elemento */
.flex-container2
.flex-item2:nth-child(2) {

    flex-shrink: 2; /* Este elemento
se encogerá el doble que el normal */
}
```

```
/* Tercer elemento */
.flex-container2
.flex-item2:nth-child(3) {
    flex-shrink: 1; /* Este elemento
se encogerá normalmente */
}
```

```
.flex-item3 {
    flex-basis: 400px; /* Cada
elemento comienza con 400px de ancho
*/
}
```

OTRAS PROPIEDADES

Ejemplos



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/MYqeWvw>

Por defecto

Item 1 Item 2 Item 3

flex-grow:1: Todos los elementos crecen de igual manera

Item 1 Item 2 Item 3

flex-shrink: Los elementos se encogen según su valor de flex-shrink cuando no caben según el valor de flex-basis

Item 1 Item 2 Item 3

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

Item 1 Item 2 Item 3

OTRAS PROPIEDADES

flex : propiedad abreviada



`flex` es una propiedad abreviada que combina `flex-grow`, `flex-shrink` y `flex-basis`.

```
.flex-item {  
    flex: 1 0 100px;  
/* flex-grow: 1, flex-shrink: 0, flex-basis: 100px */  
}
```

OTRAS PROPIEDADES

order



`order` cambia el orden en el que aparecen los elementos flexibles.

```
.flex-item {  
    order: 2;  
    /* Pone este elemento en la segunda posición */  
}
```

Los elementos con un valor menor de `order` se colocan antes. Por defecto, todos los elementos tienen `order: 0`.

RESUMEN DE PROPIEDADES COMUNES



Veamos un resumen de algunas de las propiedades más utilizadas.

Propiedad	Descripción	Valores Ejemplo
<code>display</code>	Define el contenedor como un flex container	flex
<code>flex-wrap</code>	Controla si los elementos flex deben trasladarse a la siguiente fila	nowrap (por defecto) wrap wrap-reverse
<code>justify-content</code>	Alineación horizontal de los elementos de un contenedor flexible	flex-start (por defecto) flex-end center space-between space-around space-evenly
<code>align-items</code>	Alineación vertical de los elementos de un contenedor flexible	stretch (por defecto) flex-start flex-end center baseline

RESUMEN DE PROPIEDADES COMUNES



Propiedad	Descripción	Valores Ejemplo
<code>flex-direction</code>	Define el contenedor como un flex container	row (por defecto) row-reverse column column-reverse
<code>flex-grow</code>	Controla si los elementos flex deben trasladarse a la siguiente fila	Número (valor relativo)
<code>flex-shrink</code>	Alineación horizontal de los elementos de un contenedor flexible	Número (valor relativo)
<code>flex-basis</code>	Alineación vertical de los elementos de un contenedor flexible	Valor auto (por defecto)
<code>order</code>	Visibilidad de los bordes de celdas sin contenido	Número (entero)

EJERCICIO PROPUESTO I



Crea mediante *Flexbox* tres contenedores que se encuentren inicialmente en línea y que se vayan situando unos bajo de los otros según el ancho del dispositivo.

Contenedores flexibles



Lorem ipsum dolor sit amet
consectetur adipisicing elit. Adipisci
incididunt, laudantium nostrum minus
fuga aut dolore fugit cupiditate qui.




Vivamus sagittis lacus vel augue
laoreet rutrum faucibus dolor auctor.
Integer posuere erat a ante venenatis
dapibus posuere velit aliquet.




Donec sed odio dui. Cras mattis
consectetur purus sit amet
fermentum. Nullam id dolor id nibh
ultricies vehicula ut id elit.

Contenedores flexibles



Lorem ipsum dolor sit amet
consectetur adipisicing elit. Adipisci
incididunt, laudantium nostrum minus
fuga aut dolore fugit cupiditate qui.



Vivamus sagittis lacus vel augue
laoreet rutrum faucibus dolor auctor.
Integer posuere erat a ante venenatis
dapibus posuere velit aliquet.

EJERCICIO PROPUESTO I

Solución



Código HTML:

```
<!doctype html>
<html>
  <head>
    <script
src="https://kit.fontawesome.com/509d
eda241.js"
crossorigin="anonymous"></script>
  </head>
```

```
<body>
  <div class="info-box">
    <h2>Contenedores flexibles</h2>
    <div id="flex-p">
      <p class="big"><i class="fas
fa-star"></i><br>Lorem ipsum</p>
      <p class="big"><i class="fas
fa-thumbs-up"></i><br>Lorem ipsum</p>
      <p class="big"><i class="fas
fa-check"></i><br>Lorem ipsum</p>
    </div>
  </div>
</body>
```

EJERCICIO PROPUESTO I

Solución



Código CSS:

```
*{
    margin: 0px;
    padding: 0 px;
    font-family: "Roboto", Arial,
Tahoma, sans-serif;
}
.info-box{
    background-color: #1C1D22;
    margin: 0px auto;
    padding: 110px 0;
    color: white;
    text-align: center;
}
```

```
.info-box h2{
    margin-bottom: 30px;
}
#flex-p{
    justify-content: center;
    display: flex;
    flex-wrap: wrap;
    align-items: center;
}
#flex-p p{
    padding: 30px;
    max-width: 300px;
}
```

EJERCICIO PROPUESTO I

Solución



Código CSS:

```
p.big{
  font-size: 1.1em;
  line-height: 1.5em;
}

.fa-star, .fa-thumbs-up, .fa-check{
  font-size: 2.7em;
  display: block;
  color: white;
  margin-bottom: 30px;
}
```

La salida del código anterior puede verse en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/dPbXPbq>

EJERCICIO PROPUESTO II



Crea una página web que contenga una barra de navegación horizontal en la parte superior, un pie de página en la parte inferior, y un área de contenido principal que tenga una sección principal y una barra lateral. Utiliza *Flexbox* para hacer que la barra lateral esté siempre a la derecha de la sección principal y para que la barra de navegación y el pie de página se extiendan a lo largo del ancho de la ventana del navegador.

[Inicio](#) [Acerca](#) [Servicios](#) [Contacto](#)

Título del Contenido Principal

Este es el contenido principal de la página.

Barra Lateral

Contenido de la barra lateral.

Derechos reservados © 2023

EJERCICIO PROPUESTO II

Solución



Código HTML:

```
<header class="navbar">
  <nav>
    <ul>
      <li><a href="#">Inicio</a></li>
      <li><a href="#">Acerca</a></li>
      <li><a
href="#">Servicios</a></li>
      <li><a
href="#">Contacto</a></li>
    </ul>
  </nav>
</header>
```

```
<main class="content-area">
  <section class="main-content">
    <h1>Título del Contenido
Principal</h1>
    <p>Este es el contenido principal de
la página.</p>
  </section>
  <aside class="sidebar">
    <h2>Barra Lateral</h2>
    <p>Contenido de la barra lateral.</p>
  </aside>
</main>
<footer class="footer">
  <p>Derechos reservados © 2023</p>
</footer>
```

EJERCICIO PROPUESTO II

Solución



Código CSS:

```
* {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

body {
    display: flex;
    flex-direction: column;
    /* min-height: 100vh; */
}

.navbar {
    background-color: #333;
    color: white;
    padding: 1em;
}

nav ul {
    list-style: none;
    display: flex;
}

.navbar nav a {
    color: white;
    text-decoration: none;
    padding: 0.5em;
}
```


EJERCICIO PROPUESTO II

Solución



```
.content-area {
  display: flex;
  flex-grow: 1;
  padding: 1em;
}
.main-content {
  flex-grow: 3;
  padding: 1em;
  background-color: #f8f8f8;
}
.sidebar {
  flex-grow: 1;
  padding: 1em;
  background-color: #e7e7e7;
}
```

```
.footer {
  background-color: #333;
  color: white;
  padding: 1em;
  text-align: center;
}

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

EJERCICIO PROPUESTO II

Solución



La salida del código anterior puede verse en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/wBwWBMB>

EJERCICIO PROPUESTO III



Crea un encabezado que tenga un elemento a la izquierda (texto o imagen) y otro a la derecha (menú de navegación).

Nombre del Sitio

Inicio Acerca Servicios Contacto

Dos contenedores alineados uno a la derecha y otro a la izquierda con justify-content: space-between
Menú de navegación con los elementos de tipo flex para alineación horizontal uno detrás del otro.

Pista: utiliza un contenedor principal <header> y distribuye el espacio de los elementos contenidos con justify-content. Los elementos del menú de navegación se establecen de tipo flex para alineación horizontal uno detrás del otro.

EJERCICIO PROPUESTO III

Solución



Código HTML:

```
<header>
  <h1>Nombre del Sitio</h1>
  <nav>
    <ul>
      <li><a href="#">Inicio</a></li>
      <li><a href="#">Acerca</a></li>
      <li><a
href="#">Servicios</a></li>
      <li><a href="#">Contacto</a></li>
    </ul>
  </nav>
</header>
```

```
<br>
```

```
<p>Dos contenedores alineados uno a la
derecha y otro a la izquierda con
justify-content: space-between </p>
```

```
<p>Menú de navegación con los elementos
de tipo flex para alineación horizontal
uno detrás del otro. </p>
```

EJERCICIO PROPUESTO III

Solución



Código CSS:

```
* {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
  font-size: 16px;
}
header {
  background-color: #3498db;
  color: #fff;
  padding: 15px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

```
h1 {
  margin: 0;
}
nav ul {
  list-style: none;
  display: flex;
}
nav li {
  margin-right: 20px;
}
a {
  text-decoration: none;
  color: #fff;
}
```

EJERCICIO PROPUESTO III

Solución



La salida del código anterior puede verse en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/EaYyZdm>

EJERCICIO PROPUESTO IV



Crea un encabezado que tenga un elemento arriba (texto o imagen) y otro abajo (menú de navegación). Ambos elementos deben encontrarse centrados.

Nombre del Sitio

Inicio Acerca Servicios Contacto

Dos contenedores alineados uno a la derecha y otro a la izquierda con justify-content: space-between
Menú de navegación con los elementos de tipo flex para alineación horizontal uno detrás del otro.

Pista: utiliza un contenedor principal <header> y alinean verticalmente sus elementos hijos con flex-direction. Los elementos del menú de navegación se establecen de tipo flex para alineación horizontal uno detrás del otro.

EJERCICIO PROPUESTO IV

Solución



Código HTML:

```
<header>
  <h1>Nombre del Sitio</h1>
  <nav>
    <ul>
      <li><a href="#">Inicio</a></li>
      <li><a href="#">Acerca</a></li>
      <li><a
href="#">Servicios</a></li>
      <li><a href="#">Contacto</a></li>
    </ul>
  </nav>
</header>
```

```
<br>
```

```
<p>Dos contenedores alineados
verticalmente</p>
```

```
<p>Menú de navegación con los elementos
de tipo flex para alineación horizontal
uno detrás del otro. </p>
```


EJERCICIO PROPUESTO IV

Solución



Código CSS:

```
* {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
  font-size: 16px;
}
header {
  background-color: #3498db;
  color: #fff;
  padding: 15px;
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

```
h1 {
  margin: 0;
  padding-bottom: 20px;
}
nav ul {
  list-style: none;
  display: flex;
}
nav li {
  margin-right: 20px;
}
a {
  text-decoration: none;
  color: #fff;
}
```

EJERCICIO PROPUESTO IV

Solución



La salida del código anterior puede verse en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/dPbXNwe>

EJERCICIO PROPUESTO V



Crea un menú de navegación que ocupe el ancho total de la pantalla y que sus elementos crezcan para ocupar de forma uniforme el ancho total.

Nombre del Sitio



Dos contenedores alineados verticalmente

Menú de navegación con los elementos de tipo flex para alineación horizontal uno detrás del otro. Como tiene elementos li y dentro elementos a. Primero ocupamos el espacio disponible con flex-grow:1 en los li. Después hacemos que los enlaces ocupen todo su contenedor padre haciendo display:block

Pista: utiliza la propiedad flex-grow para que los elementos del menú crezcan hasta ocupar el ancho total del menú de navegación.

EJERCICIO PROPUESTO V

Solución



Código HTML:

```
<header>
  <h1>Nombre del Sitio</h1>
  <nav>
    <ul>
      <li><a href="#">Inicio</a></li>
      <li><a href="#">Acerca</a></li>
      <li><a
href="#">Servicios</a></li>
      <li><a href="#">Contacto</a></li>
    </ul>
  </nav>
</header>
```

```
<br>
<p>Dos contenedores alineados
verticalmente</p>
<p>Menú de navegación con los elementos
de tipo flex para alineación horizontal
uno detrás del otro. Como tiene
elementos li y dentro elementos a.
Primero ocupamos el espacio disponible
con flex-grow:1 en los li. Después
hacemos que los enlaces ocupen todo su
contenedor padre haciendo display:block
</p>
```

EJERCICIO PROPUESTO V

Solución



Código CSS:

```
* {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
  font-size: 16px;
}
header {
  background-color: #3498db;
  color: #fff;
  padding: 15px;
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

```
h1 {
  margin: 0;
  padding-bottom: 20px;
}
nav ul {
  list-style: none;
  display: flex;
}
nav li {
  margin-right: 20px;
}
a {
  text-decoration: none;
  color: #fff;
}
```

EJERCICIO PROPUESTO V

Solución



La salida del código anterior puede verse en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/RNbRKve>