



Tabla de contenido

Flexbox	1
Notas	3
Filas en lugar de columnas	3
Orden de los elementos: order	4
Crecimiento de los elementos : flex-grow	4
Decrecimiento de los elementos : flex-shrink	5
Espacio entre elementos: gap, column-gap y row-gap	5
Propiedad flex-wrap	5
Propiedad flex-flow	5
Propiedad justify-content	6
Propiedad align-content	7
Propiedad align-items	8
Propiedad flex-basis	9
Propiedad flex	9
Propiedad align-self	9

Flexbox

Es un módulo de CSS creado para poder diseñar, de manera más fácil y con muchas más posibilidades, nuestras páginas, en lo relativo a filas y columnas, además de ofrecer muchas opciones para reordenar y recolocar su contenido; lo cual es muy útil para un diseño responsivo.

Para ello debemos crear un elemento contenedor con la propiedad **display** a los valores **flex** (tipo bloque) o **inline-flex** (en línea). Luego dentro colocaremos las filas y columnas deseadas usando otras propiedades que iremos viendo. Inicialmente se usaba el valor **flexbox** en lugar de *flex*.

Ejemplo:

El CSS es:

```
.contenedor {
  border:1px solid black;
  display: flex;
}
.caja {
  color: white;
  font-family: Arial, Helvetica, sans-serif;
  padding: 5px;
  border: 1px solid black;
}
.caja1 {
  background-color: rgb(24, 135, 226); /* Un azul */
}
.caja2 {
  background-color: rgb(43, 230, 186); /* Un verde turquesa */
}
```

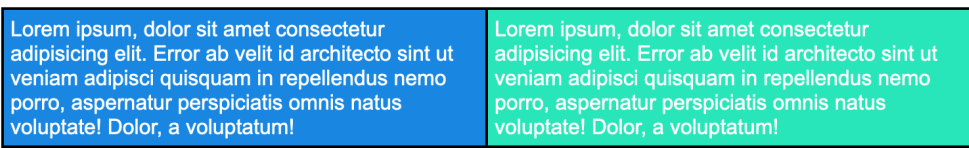


CSS - Diseñando con Flexbox

```
}  
.caja3 {  
  background-color: rgb(223, 75, 112); /* Un rojo claro */  
}
```

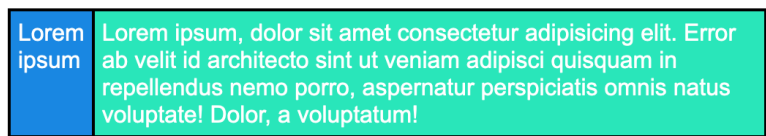
El HTML es:

```
<div class="contenedor">  
  <div class="caja caja1">Lorem ipsum, dolor sit amet consectetur adipisicing elit. Error ab velit  
  id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis  
  omnis natus voluptate! Dolor, a voluptatum!</div>  
  <div class="caja caja2">Lorem ipsum, dolor sit amet consectetur adipisicing elit. Error ab velit  
  id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis  
  omnis natus voluptate! Dolor, a voluptatum!</div>  
</div>
```

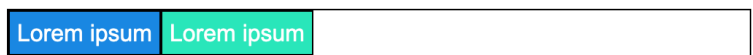


Podemos ver que se han creado dos columnas, cada una con el mismo ancho, pues ambas tienen el mismo contenido. Si cambiamos el tamaño de la ventana, las columnas crecerán o decrecerán de acuerdo con este.

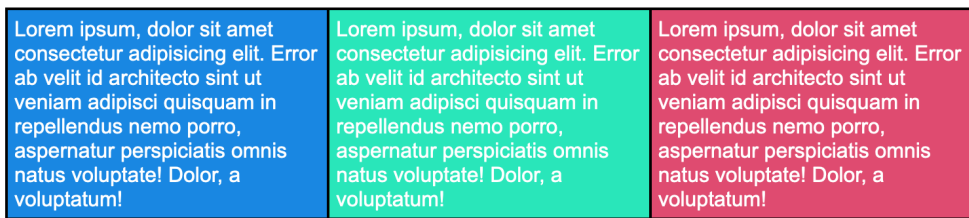
Si por ejemplo, el primer div solo tuviese el texto *Lorem ipsum*, el resultado sería:



Si ambos elementos ocupan poco, quedará libre parte del contenedor, si es de tipo bloque (flex).



Si añadimos un tercer div con la clase *caja* y *caja3*, el tamaño ahora se reparte entre las tres columnas.



Flexbox intenta que el contenido de las columnas se muestre siempre, empequeñeciendo las columnas todo lo que le permitan su contenido.

Lorem ipsum, dolor sit amet consectetur adipiscing elit. Error ab velit id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis omnis natus voluptate! Dolor, a voluptatum!	Lorem ipsum, dolor sit amet consectetur adipiscing elit. Error ab velit id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis omnis natus voluptate! Dolor, a voluptatum!	Lorem ipsum, dolor sit amet consectetur adipiscing elit. Error ab velit id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis omnis natus voluptate! Dolor, a voluptatum!
--	--	--

Empequeñeciendo mucho la ventana

Si aun así no cupiese todo, habrá parte que no se vea, debiendo desplazarnos horizontalmente por la página para verlo.

Notas

- Todos estos posicionamientos los hace teniendo en cuenta el idioma, con lo que en algunos colocará los elementos de derecha a izquierda en lugar de izquierda a derecha. Nosotros vamos a suponer siempre un idioma de izquierda a derecha y de arriba abajo.
- Por defecto las siguientes propiedades y comportamientos están establecidos a:
 - `flex-direction: row`; con lo que los elementos se colocan en una fila
 - los elementos no se estiran en el eje principal aunque haya sitio disponible, pero pueden encoger para hacer sitio
 - `align-items: stretch`; los elementos se estiran en el eje de cruce (el contrario al eje principal) para rellenar el espacio sobrante. Por ejemplo, en modo filas, si le damos un alto al contenedor y los elementos son más cortos, ocuparán todo el alto; lo mismo si hay elementos más altos que otros.
 - `flex-wrap: nowrap`; con lo que los elementos permanecen siempre en una sola fila o columna; si no caben, sobresaldrán de su contenedor
 - la propiedad `width` en los elementos no se tienen en cuenta, pero sin `min-width`
- The [flex-basis](#) property is set to `auto`. This means that, in each case, it will be equal to the flex item [width](#) in horizontal writing mode, and the flex item [height](#) in vertical writing mode. If the corresponding width/height is also set to `auto`, the `flex-basis` content value is used instead.
-
-
- En los siguientes ejemplos de este manual, salvo que se indique lo contrario, vamos a suponer que usamos los valores por defecto `flex-direction: row` y `flex-wrap: nowrap`.
- También vamos a suponer, salvo que se indique lo contrario, que los elementos no tienen establecido ningún ancho ni alto.

Filas en lugar de columnas

Basta con usar la propiedad **`flex-direction`** con el valor `column`, lo cual puede sonarnos un poco contradictorio, pues no crea varias columnas, sino filas. Esta propiedad dicta la dirección del eje principal.

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Error ab velit id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis omnis natus voluptate! Dolor, a voluptatum!

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Error ab velit id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis omnis natus voluptate! Dolor, a voluptatum!

Esta propiedad también admite los siguientes valores:

- **row**: muestra los contenidos en una sola fila. Es el valor por defecto y la que se ha usado en los ejemplos anteriores
- **row-reverse**: como row pero mostrando los contenidos en orden inverso. En nuestro ejemplo anterior, primero mostraría primero el segundo div y luego el primero
- **column-reverse**: como column pero mostrando los contenidos en orden inverso.

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Error ab velit id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis omnis natus voluptate! Dolor, a voluptatum!

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Error ab velit id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis omnis natus voluptate! Dolor, a voluptatum!

Ejemplo con row-reverse

Orden de los elementos: order

Con la propiedad **order** podemos indicar la posición que queremos donde aparezca un elemento, siendo un valor numérico, positivo o negativo que permite colocar unos elementos antes que otros; a menor valor, antes aparecerán. El valor por defecto es cero. Es una propiedad de los elementos, no del contenedor.

Por ejemplo, si a la clase caja2 le añadimos la propiedad `order:-1`, se mostrará primero. Podríamos conseguir lo mismo usando `order: 1` en la caja1, pues caja2 tendría el valor 0.

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Error ab velit id architecto sint ut veniam adipisci quisquam in repellendus nemo porro, aspernatur perspiciatis omnis natus voluptate! Dolor, a voluptatum!

Lorem ipsum

Crecimiento de los elementos : flex-grow

Si queda sitio libre en el contenedor, podemos usar la propiedad **flex-grow** de un elemento para que crezca ocupando ese espacio. Es un valor numérico, por defecto cero con lo que no crecen; a mayor valor, más crece un elemento. Por ejemplo, si tenemos un elemento con el valor 1 y el otro con 2, el segundo crecerá el doble (lo mismo si usamos 10 y 20, 100 y 200 o 0.5 y 1). Es una propiedad de los elementos, no del contenedor.

En los siguientes ejemplos suponemos que ambos div solo tienen el texto Lorem ipsum:

Lorem ipsum

Lorem ipsum

flex-grow:1 en el primero y flex-grow:2 en el segundo

Lorem ipsum

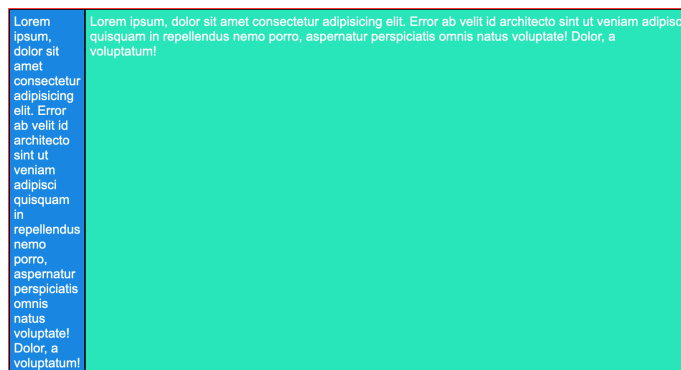
Lorem ipsum

flex-grow:0 o nada en el primero y flex-grow:1 en el segundo

Decrecimiento de los elementos : flex-shrink

Es lo contrario a flex-grow, a mayor número, menos empeequeñecerá un elemento cuando sea necesario, siendo 1 el valor por defecto, pues todos empeequeñecen si es necesario. Es una propiedad de los elementos, no del contenedor.

En el siguiente ejemplo, ponemos flex-shrink a 2 en el primer elemento:



Espacio entre elementos: gap, column-gap y row-gap

Con estas propiedades podemos establecer un margen entre columnas filas o ambas.

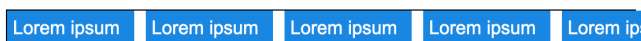
Propiedad flex-wrap

Por defecto, con el valor **nowrap**, los elementos irán en una sola fila, o columna según flex-direction, pudiendo desbordar su contenedor. Si usamos el valor **wrap**, se pasarán a la siguiente fila o columna si es necesario; aunque también podría haber desbordamiento. Con la opción **wrap-reversed**, si es necesario bajar de fila alguno, se harán de izquierda a derecha y no primero el de la derecha.

En los siguientes ejemplos vamos a suponer que tenemos 6 cajas con un min-width de 100 píxeles cada una, además un un gap de 10px.



Con el valor wrap



Con el valor nowrap

Propiedad flex-flow

Permite indicar las propiedades flex-direction y flex-wrap en una sola; primero la dirección y luego el wrap.

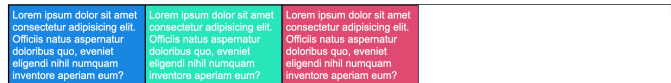
Ejemplo:

```
flex-flow: flex-column wrap;
```

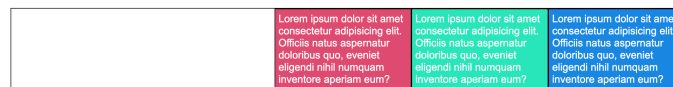
Propiedad justify-content

Si sobra espacio podemos usar la propiedad **justify-content** del contenedor para indicar donde colocar los elementos con respecto al mismo. Sus posibles valores son:

- **flex-start**: los elementos se comienzan a colocar por la parte izquierda, pudiendo quedar espacio libre por la derecha. Es el valor por defecto. Si hemos usado row-reverse, se comienzan a colocar por su parte derecha. En caso de colocar los elementos en columna, hablaremos de arriba y abajo

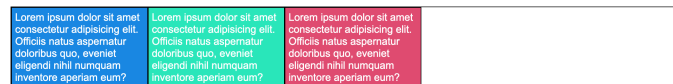


Hemos indicado un max-width de 200px en cada elemento para que no ocupen todo el ancho

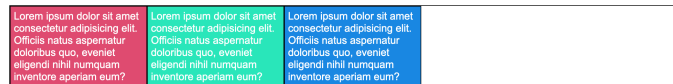


Con flex-direction: flow-reversed

- **start**: como flex-start pero no tiene en cuenta si la dirección está inversa, es decir, que siempre se colocan en la izquierda

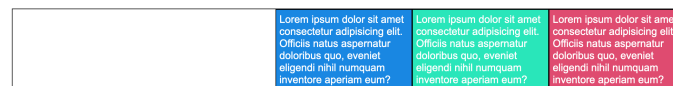


Hemos indicado un max-width de 200px en cada elemento para que no ocupen todo el ancho

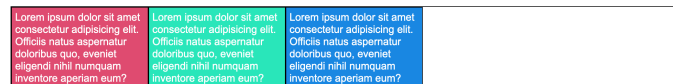


Con flex-direction: flow-reversed

- **flex-end**: los elementos se colocan en la parte derecha, pudiendo quedar espacio libre por la izquierda. En caso de colocar los elementos en columna, hablaremos de abajo a arriba

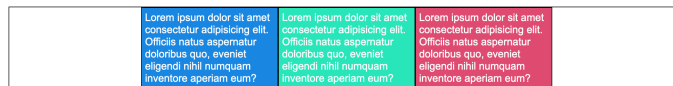


Hemos indicado un max-width de 200px en cada elemento para que no ocupen todo el ancho



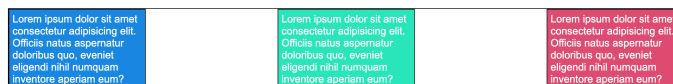
Con flex-direction: flow-reversed

- **end**: el equivalente a start para flex-end
- **center**: los elementos se colocan centrados con respecto a su contenedor, ya sea vertical u horizontalmente según la dirección, pudiendo quedar espacio libre por los lados



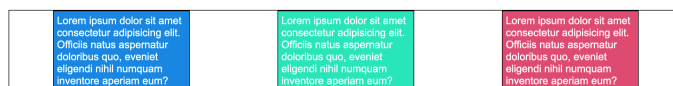
Hemos indicado un max-width de 200px en cada elemento para que no ocupen todo el ancho

- **space-between:** los elementos se distribuyen de manera uniforme con el primero y último alineados a sus esquinas



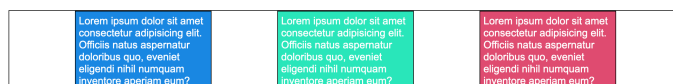
Hemos indicado un max-width de 200px en cada elemento para que no ocupen todo el ancho

- **space-around:** los elementos se distribuyen de manera uniforme, aunque veremos que los de los lados tienen menos espacio vacío, pues el central tiene la suma de su espacio y el de los otros



Hemos indicado un max-width de 200px en cada elemento para que no ocupen todo el ancho

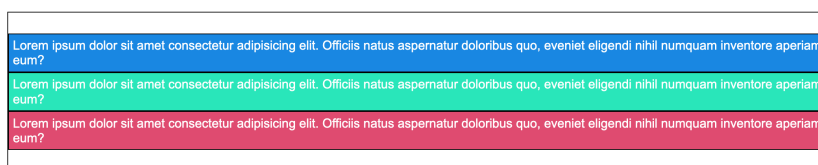
- **space-evenly:** como el anterior, pero ahora el espacio vacío se distribuye igual entre todos los elementos



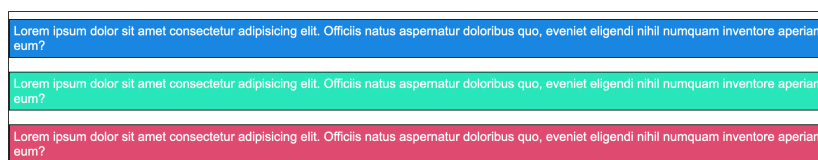
Hemos indicado un max-width de 200px en cada elemento para que no ocupen todo el ancho

Propiedad align-content

Si varios elementos no caben en la misma fila, en el caso de una dirección row, en una columna, en el caso de una dirección column y hay espacio sobrante en el contenedor, esta propiedad es similar a justify-content, añadiendo también el valor stretch para que las líneas ocupen el tamaño libre, si no lo impide su tamaño máximo.



center: hemos dado un alto al contenedor y usado wrap



space-around: hemos dado un alto al contenedor y usado wrap

Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?
Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?
Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?

stretch: hemos dado un alto al contenedor y usado wrap

Propiedad align-items

Con la propiedad **align-items** del contenedor podemos ajustar como se disponen los elementos verticalmente en el contenedor, si se muestran horizontalmente, u horizontalmente si se muestran verticalmente. Sus valores son flex-start (arriba) y start, flex-end y end (abajo), center (centrados verticalmente), stretch (ampliados para que ocupen todo el alto; es el valor por defecto) y baseline (centrados por la línea base). Esto si hablamos de una disposición en filas de arriba a abajo; la cosa cambiará cuando sea en columnas o de abajo a arriba.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?	Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?	Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?
---	---	---

center: suponiendo un alto para el contenedor y que los elementos se colocan en filas

Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?	Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?	Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?
---	---	---

flex-end: suponiendo un alto para el contenedor y que los elementos se colocan en filas

Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?	Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?	Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis natus aspernatur doloribus quo, eveniet eligendi nihil numquam inventore aperiam eum?
---	---	---

stretch: suponiendo un alto para el contenedor y que los elementos se colocan en filas

Propiedad flex-basis

Es una propiedad un poco compleja, pues sirve para dar un tamaño base a los elementos, pero este dependerá de otros factores como el espacio libre, el de los otros elementos, las propiedades flex-grow, flex-shrink, ...

Lorem ipsum dolor sit, amet consectetur adipiscing elit. Maiores, error?	Lorem ipsum dolor sit amet consectetur adipiscing elit. Commodi aspernatur veniam	Lorem ipsum dolor sit, amet consectetur adipiscing elit. Tenetur sint, quos voluptas	
--	---	--	--

0 sin anchos ni anchos mínimos

Lorem ipsum dolor sit, amet consectetur adipiscing elit. Maiores, error?	Lorem ipsum dolor sit amet consectetur adipiscing elit. Commodi aspernatur veniam corporis aut nesciunt fuga ut ad quod minus nisi!	Lorem ipsum dolor sit, amet consectetur adipiscing elit. Tenetur sint, quos voluptas possimus fugiat dolorem.
--	---	---

auto sin anchos ni anchos mínimos

Propiedad flex

Es una combinación de *grow shrink basis*); en ese orden. También se puede poner auto (1 1 auto), initial (0 1 auto), none (0 0 auto). Por defecto 0 1 auto.

Propiedad align-self

Permite cambiar la posición de un elemento en concreto cuando hemos usado la propiedad *align-items* en el contenedor.

	Lorem ipsum dolor sit amet consectetur adipiscing elit. Commodi aspernatur veniam corporis aut nesciunt fuga ut ad quod minus nisi!	Lorem ipsum dolor sit, amet consectetur adipiscing elit. Tenetur sint, quos voluptas possimus fugiat dolorem.
Lorem ipsum dolor sit, amet consectetur adipiscing elit. Maiores, error?		

align-items a flex-start en el contenedor y align-self a flex-end en el primer div