

# el modelo flexbox

Antes había cuatro modos de disposición de los espacios (layout):

**block**, por secciones en una página web

**inline**, para el texto (hacks de clearfix)

**table**, para datos de tabla de dos dimensiones (<table> y display: table hacks)

**position**, para la posición explícita de un elemento (floats)

**flexbox** es un modelo de disposición unidimensional, esto significa que va a controlar el layout basado en la disposición de los elementos en la columna (column) o en la fila (row), pero no a la vez en ambos.

El principal objetivo de flexbox es permitir a los elementos llenar todo el espacio ofrecido por el contenedor dependiendo de las reglas que le seteamos.

*La diferencia más grande entre Bootstrap 3 y Bootstrap 4 es que la última versión utiliza flexbox, en lugar de float para manejar las disposición de los elementos.*

## Aplicar Flexbox

flexbox se aplica a un contenedor, colocando la regla:

`display: flex;` o `display: inline-flex;`

El contenido (content) dentro del contenedor (container) será alineado usando flexbox.

`display: inline-flex` no hace que los items flex se muestren inline. Hace que el contenedor se disponga inline con respecto a otros contenedores.

Esta es la única diferencia entre `display: inline-flex` and `display: flex`.

Algo similar a lo que sucede entre `display: inline-block` y `display: block`.

No hay ninguna diferencia sobre los items flex dentro del contenedor.

## Propiedades del Container

Estas propiedades aplicadas sobre el container, establece las reglas generales para los items contenidos.

`flex-direction`, `justify-content`, `align-items`, `flex-wrap` y `flex-flow`

### flex-direction

Determina si los ítems se alinean en una fila (row) o en una columna (column):

`flex-direction: row;`

coloca los ítems en una fila, en la dirección del texto (izquierda a derecha)

`flex-direction: row-reverse;`

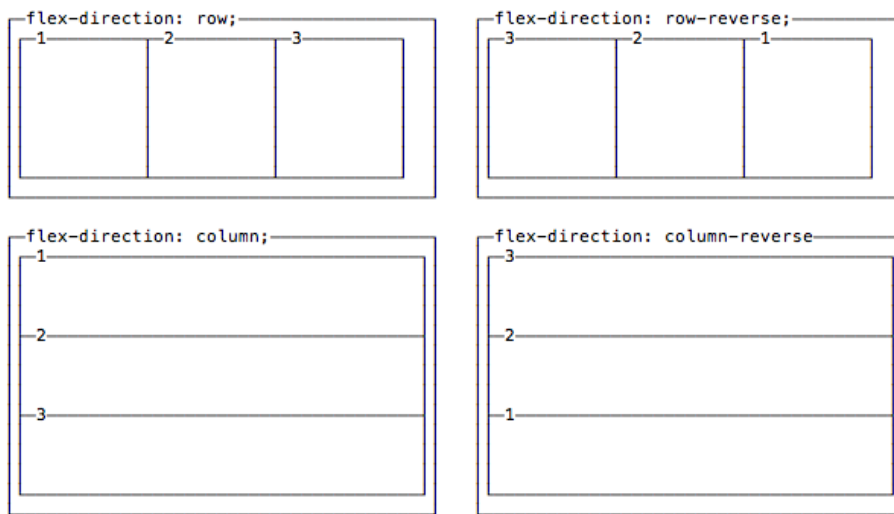
coloca los ítems también en una fila, pero en la dirección opuesta (derecha a izquierda)

`flex-direction: column;`

coloca los ítems en una columna, ordenándolos de arriba hacia abajo

`flex-direction: column-reverse;`

coloca los ítems en una columna, pero ordenándolos de manera inversa (de abajo hacia arriba)



Predeterminadamente (by default) los ítems comienzan desde la izquierda si flex-direction es row, y desde arriba si flex-direction es column.

## justify-content

ajusta el alineamiento horizontal del contenido

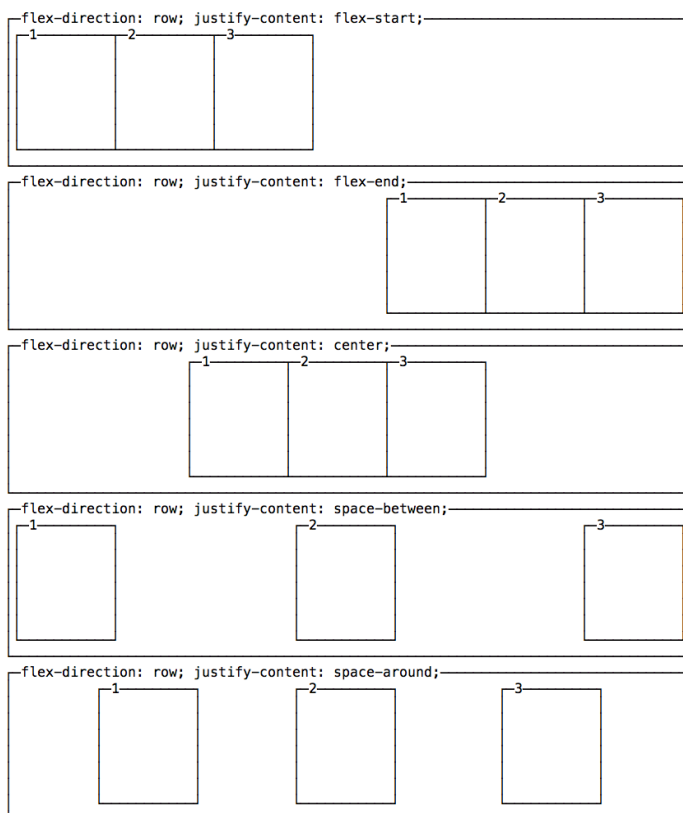
**flex-start:** alinear los elementos hacia la izquierda del contenedor.

**flex-end:** alinear los elementos hacia la derecha del contenedor.

**center:** alinear los elementos con el centro del contenedor.

**space-between:** despliega los elementos con igual espacio de separación entre ellos.

**space-around:** despliega los elementos con igual espacio alrededor de cada uno.



## align-items

cambia el alineamiento vertical de los elementos y tiene 5 posibles valores:

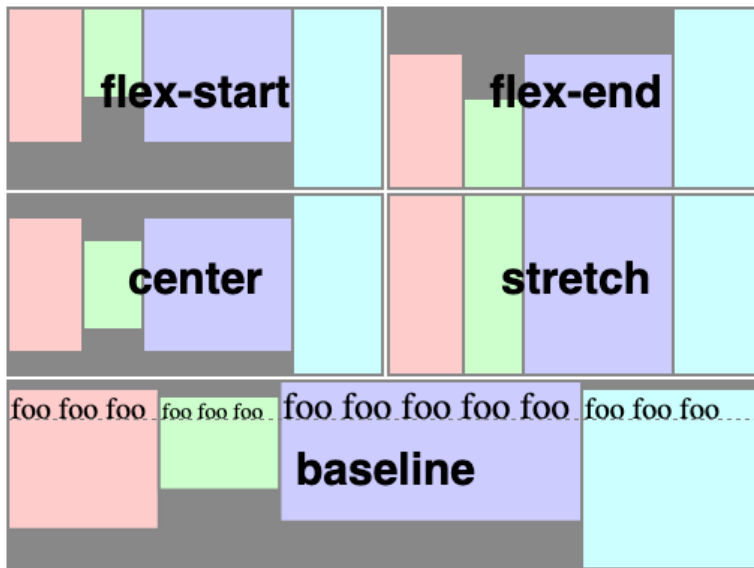
**flex-start**: alinear los elementos a la parte superior del contenedor.

**flex-end**: alinear los elementos a la parte inferior del contenedor.

**center**: alinear los elementos con el centro vertical del contenedor.

**baseline**: despliega los elementos alineados por la línea de base (baseline) de cada uno de los ítems del contenedor.

**stretch**: los ítems se estiran para adaptarse al contenedor.



## flex-wrap

flex-wrap especifica si los elementos "hijos" son obligados a permanecer en una misma línea o pueden fluir en varias líneas.

**flex-wrap: nowrap;**

valor predeterminado los ítems en un contenedor flexbox son mantenidos en una única línea, ajustándose al contenedor.

**flex-wrap: wrap;**

Los elementos flex son colocados en varias líneas.

**flex-wrap: wrap-reverse;**

actúa como wrap pero revierte el orden.

## atajos

### flex-flow

esta instrucción es un atajo permite especificar a la misma vez flex-direction y flex-wrap en una sola línea, por ejemplo:

**flex-flow: row wrap;**

## Propiedades que se aplican a cada ítem individualmente

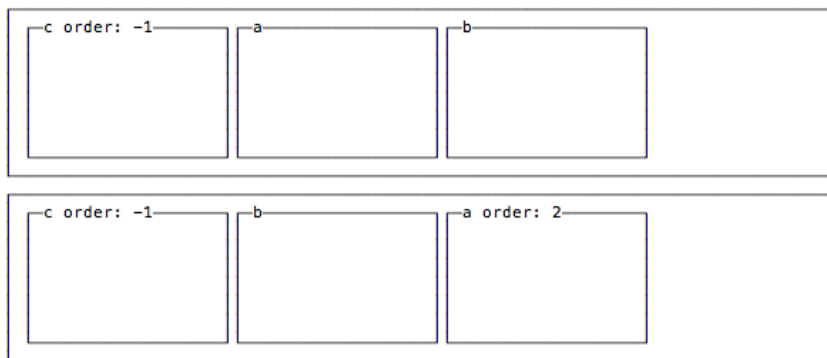
Los elementos “hijos” tienen cierto grado de independencia y flexibilidad, se puede cambiar su apariencia con las siguientes propiedades:

- order
- align-self
- flex-grow
- flex-shrink
- flex-basis
- flex

### order

los ítems son ordenados de manera predeterminada por la manera en que se escriben en el html, y el valor por default de order es 0 (cero).

Se puede asignar un valor para que ejecute un orden en particular sobre cada ítem separadamente. También podemos hacer aparecer un elemento al principio de los demás asignando un valor negativo al valor de orden del ítem.



### align-self

Por separado cada ítem con esta sentencia puede anular la instrucción align-items del contenedor usando align-self, el cual tiene 5 valores posibles:

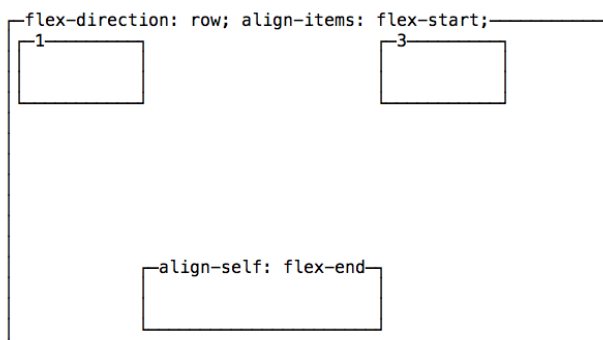
flex-start: se alinea a la parte superior del contenedor.

flex-end: se alinea a la base del contenedor.

center: se posiciona alineado verticalmente con el centro del contenedor.

baseline: se muestra alineado con la línea base del contenedor (baseline).

stretch: el ítem se adapta para llenar el total del lado del contenedor.



## flex-grow

el valor predeterminado para cada elemento es 0 (cero)

Si todos los ítems se definen como 1 y uno de ellos se define como 2, el elemento con el valor mayor asignado (2), tomará el espacio correspondiente a 2 ítems con valor 1. (expandir = grow)

## flex-shrink

el valor predeterminado para cualquier elemento es 1 (uno)

Si todos los ítems están definidos como 1 y uno se define como 3, este elemento se encoge (shrink) 3 veces en comparación a los elementos con valor 1.

## flex-basis

el valor predeterminado es **auto**, esto significa que el ítem se adapta acorde al ancho o alto, y agrega extra espacio basado en la propiedad flex-grow

si el valor es 0 (cero), no agrega ningún espacio extra para el ítem.

si se especifica un valor en pixeles, usara ese valor como tamaño (ancho o alto según sea un item row o column)

estas son algunas de las posibilidades en resumen, para mas informacion sobre las posibilidades de flex-basis consultar: <https://developer.mozilla.org/es/docs/Web/CSS/flex-basis>

## flex






Esta propiedad combina las tres propiedades anteriores:

- flex-grow
- flex-shrink
- flex-basis

y brinda un atajo de sintaxis, por ejemplo: `flex: 0 1 auto;`

## Soporte del navegador

Las propiedades flexbox son compatibles con todos los navegadores modernos.

				
29.0	11.0	22.0	10	48

## Referencias

- <https://flaviocopes.com/flexbox/>
- <https://developer.mozilla.org/es/docs/Web/CSS/flex-wrap>
- <http://w3.unpocodetodo.info/css3/ejemplos/flexbox-playground.html>