Flexbox:

Básicamente, Flexbox propone una manera de organizar los elementos presentes en nuestra estructura de HTML. Busca ser mucho más orgánica y fluida, pensada en los comportamientos naturales que posee cualquier elemento de la vida real con cuatros costados (arriba, derecha, abajo, izquierda).

Propone un **único flujo**, en el que dispondremos de los elementos con mayor libertad para **distribuir**, **redimensionar** y **reordenar** cada uno de ellos en función de ese flujo de trabajo.

¿Qué es un eje?

los ejes dentro de un contenedor flex definen la orientación a partir de la cual se desplazarán los elementos internos del mismo.

Un contenedor flex posee dos ejes: el eje principal, llamado también main axis, y el eje transversal, llamado cross axis. En función de cuál es el eje principal, los elementos se distribuyen en **filas horizontales** o en **columnas verticales**.

**flex-direction**

Con esta propiedad **definimos** el **main axis** (eje principal) del contenedor, que puede ser tanto **horizontal** como **vertical**. El **cross axis** (eje transversal) será la dirección perpendicular al main axis.

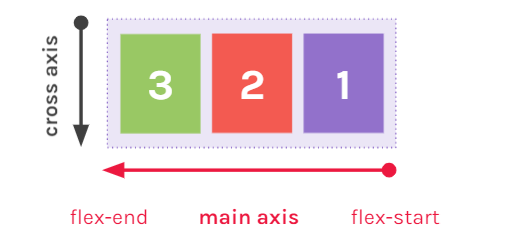
**flex-direction: row**

Los ítems se disponen en el **eje x**, de **izquierda** a **derecha**. Si no le aclaramos la propiedad flex-direction al contenedor, rowes el **valor por defecto**.

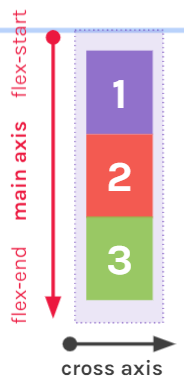


**flex-direction: row-reverse**

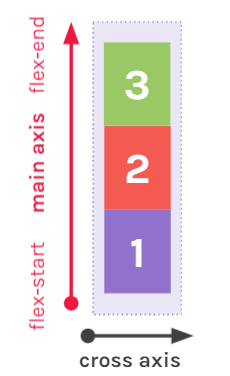
Los ítems se disponen en el **eje x**, de **derecha** a **izquierda**. En este caso, estamos invirtiendo el inicio y fin del **main-axis**.



**flex-direction: columna**: Los ítems se disponen en el **eje y**, de **arriba** hacia **abajo**.

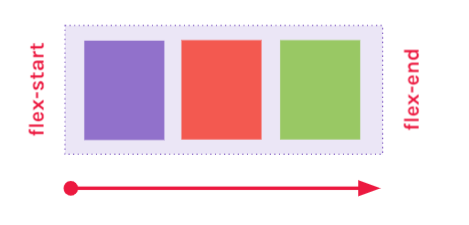


**flex-direction: column-reverse:** Los ítems se disponen en el **eje y**, de **abajo** hacia **arriba**.En este caso, estamos invirtiendo el inicio y fin del **main-axis**.

****

**justify-content**

Con esta propiedad **alineamos** los **ítems** a lo largo del **main axis**. Si es horizontal, se alinearán en función de la fila. Si es vertical, se alinearán en función de la columna.

****

flex-start es el **valor por defecto** del Justify

**justify-content: flex-end**

Los ítems se alinean respecto del **final** del **main axis** que hayamos definido.

**justify-content: center**

Los ítems se alinean en el **centro** del main axis.

**justify-content: space-between**

Los ítems se **distribuyen** de manera **uniforme**. El **primer ítem** será enviado al **inicio** del **main axis**, y el **último ítem**,al **final**. El espacio libre se repartirá para separar los ítems.

**justify-content: space-around**

Los ítems se **distribuyen** de manera **uniforme**. El espacio libre disponible se repartirá entre todos los elementos. Del espacio que le toque a cada elemento, la mitad irá a la derecha y la otra a la izquierda (o arriba y abajo en caso de que sean columnas).

**align-items**

Con esta propiedad **alineamos** los **ítems** a lo largo del **cross axis**. Si no aclaramos esta propiedad, el **valor por defecto es stretch**, en otras palabras, los ítems ocuparán todo el espacio disponible en el **cross axis**.

**align-items: stretch**

Los **ítems** se **ajustan** para abarcar **todo** el **contenedor**. Si el **cross axis** es **vertical**, se ajustan en función de la **columna**. Si el **cross axis** es **horizontal**, se ajustan en función de la **fila**.

**align-items: flex-start**

Los **ítems** se **alinean** al **inicio** del **cross-axis**.

**align-items: flex-end**

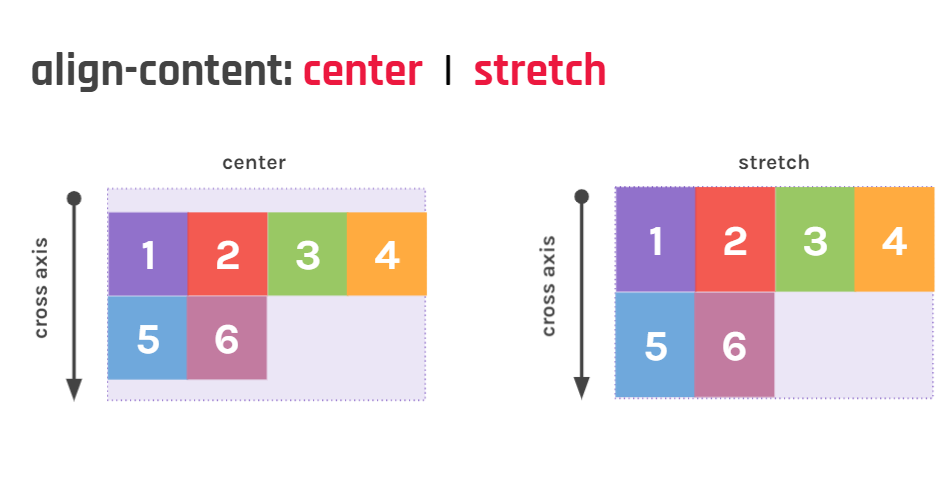
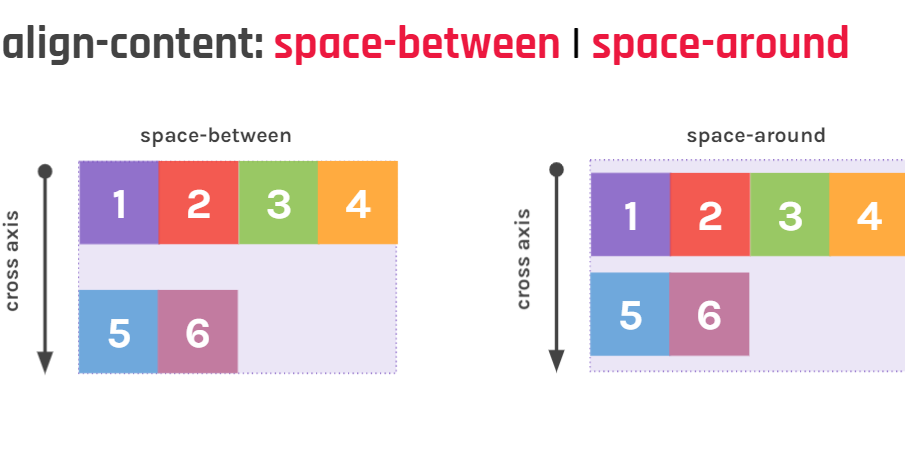
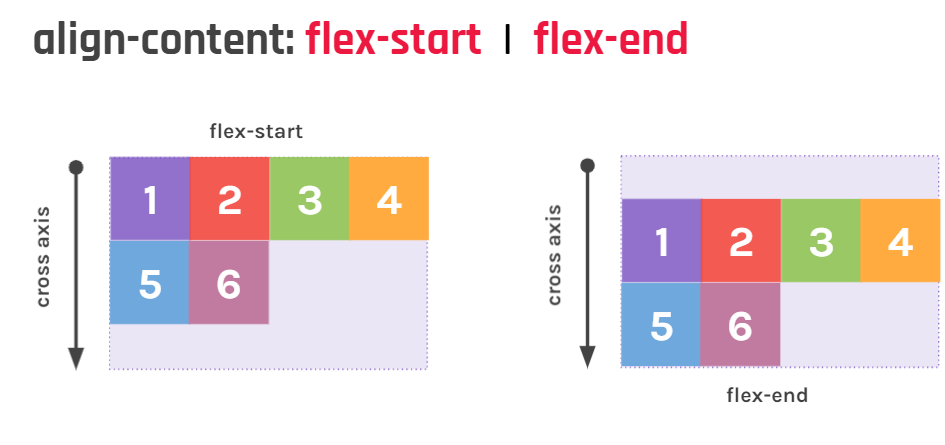
Los **ítems** se **alinean** al **final** del eje transversal.

**align-items: center**

Los **ítems** se **alinean** al **centro** del eje transversal.

**align-content**

En caso de que estemos trabajando con contenedores **multilínea** Con esta propiedad **alineamos** los **ítems** a lo largo del **cross axis** cuando los contenedores flexibles incluyen de varias líneas (donde **flex-flow** se establece en **wrap** o **wrap-reverse**).

****

Estructura de Flexbox:

**Flexbox** propone una estructura basada en el uso de un **contenedor padre** (*flex-container*) y sus **elementos hijos** (*flex-items*).

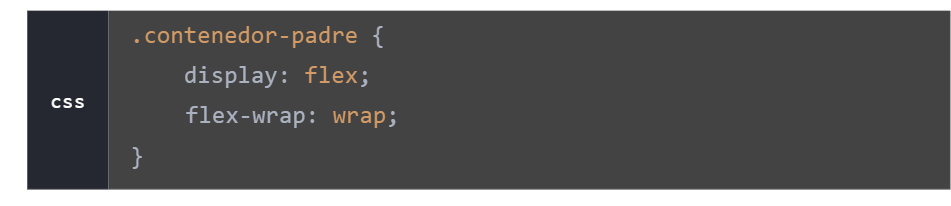


Cuando hablamos de un flex-container, hablamos de un elemento HTML que contiene a uno o más elementos. A estos elementos anidados los llamamos flex-items. En el flex-container es en donde configuramos la mayoría de las propiedades flex.

**flex-wrap**

Por defecto, los elementos hijos de un contenedor flex van a tratar de **entrar** todos en una **misma línea**.

Para aclararle al contenedor que **debe respetar** el **ancho** definido de sus **hijos** usamos la propiedad flex-wrapcon el valor wrap.



Items

***order***

Con esta propiedad **controlamos** el **orden** de cada ítem, sin importar el orden original que tengan en la estructura HTML. Esta propiedad recibe un **número entero**, **positivo o negativo**,como **valor**. Por defecto, todos los ítems flex tienen un order: 0 implícito, aunque no se especifique.

**order: número positivo**

Si le asignamos a la **caja Q** (que posee la clase caja-q) la propiedad order con valor 1, esta **pasará al final** de la fila por ser el **número más alto**. Recordemos que, por defecto, el valor del orden de cada ítem es 0.

**order: número negativo**

Si ahora le asignamos a la **caja D** la propiedad order con un -1como valor, esta pasará al **principio** de la fila. Colocando al ítem con el orden más pequeño primero.

Las cajas se irán **ordenando respetando** la **secuencia** desde  
los números **negativos** hacia los **positivos**.

***flex-grow***

Con esta propiedad definimos cuánto puede llegar a **crecer** un **ítem** en caso de **disponer** de **espacio libre** en el contenedor. Configura un crecimiento flexible para el elemento. Si **ambos ítems** tienen la propiedad flex-grow con valor 1, a medida que el contenedor se agrande, irán abarcando el espacio disponible en partes iguales.

Si un solo **ítem** tienen la propiedad flex-grow, este intentará ocupar el espacio libre disponible, a medida que el contenedor se agrande, según la proporción que definamos con el valor.

***align-self***

Nos permite **alinear**,sobre el **cross axis**,a cada ítemal que le **apliquemos** esta propiedad, independientemente de la **alineación** que se haya definido en el **contenedor flex** con **align-items**.

**align-self: flex-end**

**align-self: center**

**align-self: flex-start**

estos 3 se ainean con respecto al eje transversal

**align-self: stretch en cambio este** Con stretch, el ítem se **ajusta** hasta abarcar **todo** el **cross axis**, es el comportamiento por defecto.

Estas **propiedades aplicarán** para los flex-items siempre y cuandoel contenedor **padre** sea un **flex-container**.