

# Ejes

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree

# Índice

1. [Ejes en Flexbox](#)
2. [flex-direction](#)
3. [justify-content](#)
4. [align-items](#)
5. [align-content](#)

1

# Ejes en Flexbox



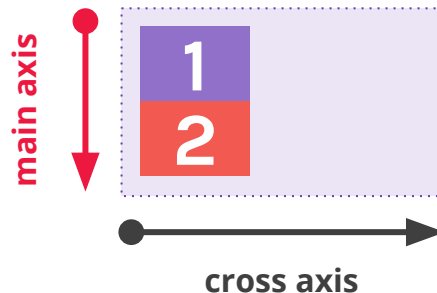
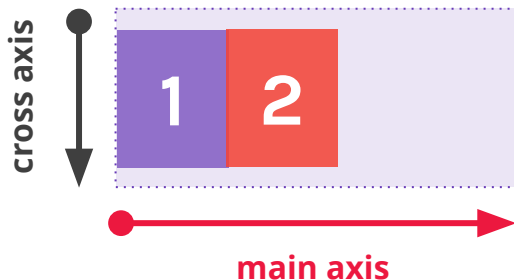
Flexbox trabaja con dos ejes para **desarrollar** todo su **flujo** interno: el **eje X** y el **eje Y**.

Según cómo decidamos ordenar los elementos, llamaremos **main axis** a uno y **cross axis** al otro.



# Ejes en Flexbox

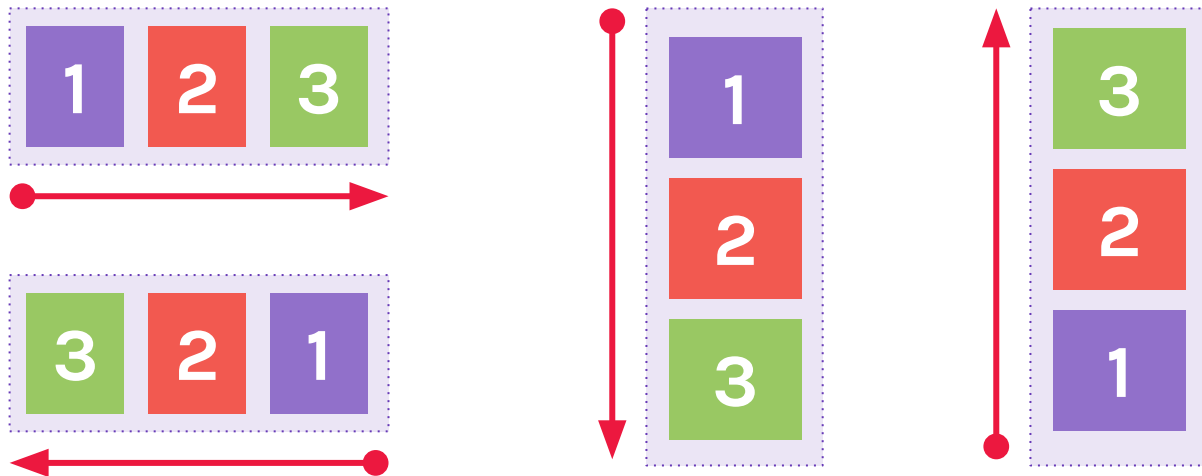
Cuando trabajamos en un flujo flex, hablamos del **main axis** y el **cross axis**. Definiendo el **eje principal** de nuestro contenedor flex estamos determinando el flujo que tendrán los elementos dentro del contenedor. En función de cuál es el eje principal, los elementos se distribuyen en **filas horizontales** o en **columnas verticales**.



## 2 | flex-direction

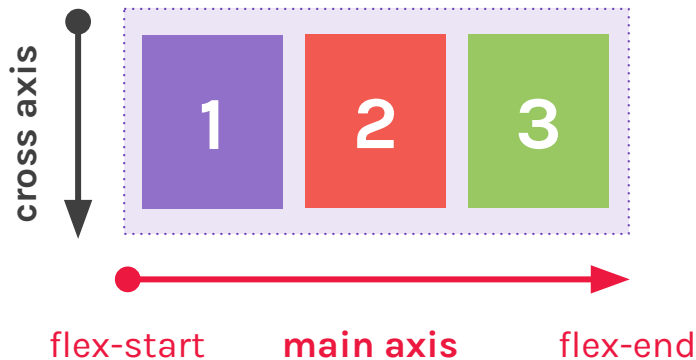
# flex-direction

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

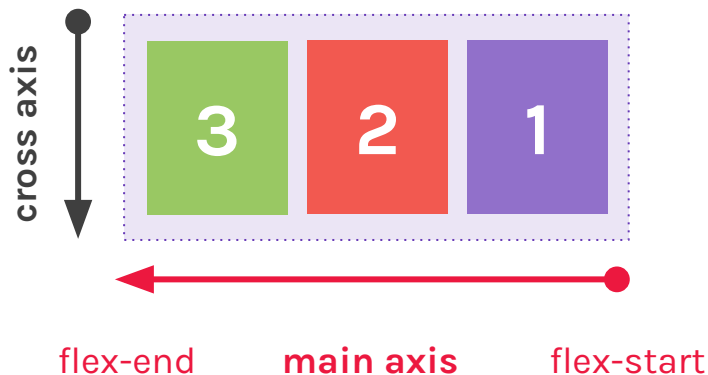
Los ítems se disponen en el **eje x**, de **izquierda** a **derecha**. Si no le aclaramos la propiedad `flex-direction` al contenedor, `row` es el **valor por defecto**.





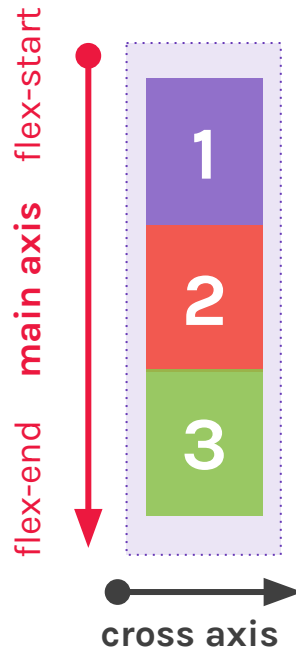
## flex-direction: row-reverse

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: column

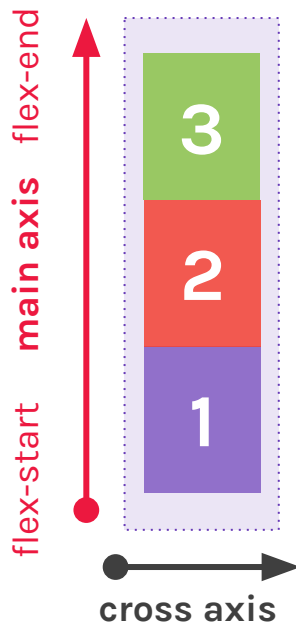
Los ítems se disponen en el **eje y**, de **arriba** hacia **abajo**.



# 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**.





Flexbox nos da **dos propiedades** para **alinear** fácilmente los elementos.

A través del **main axis** con **justify-content**.

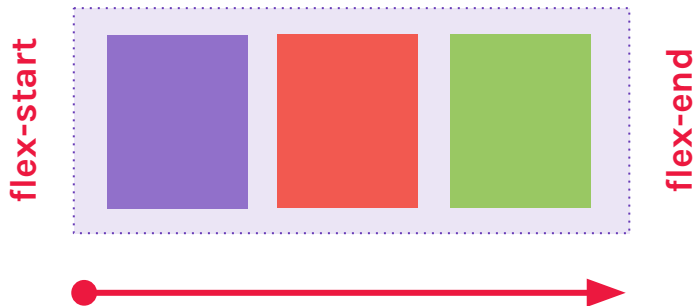
A través del **cross axis** con **align-items**.



# 3 | justify-content

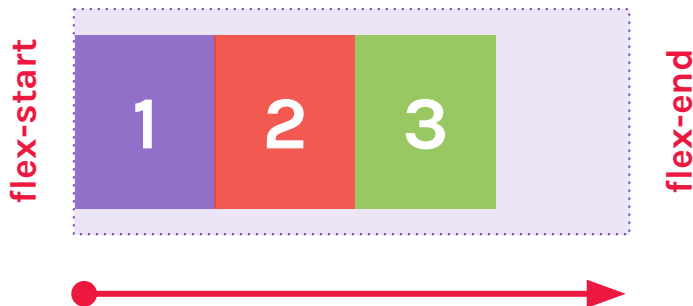
# justify-content

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.



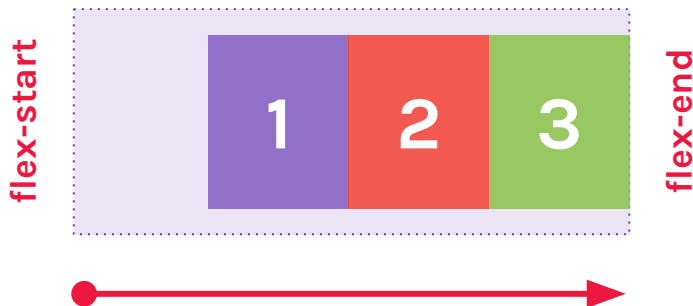
# justify-content: flex-start

Los ítems se alinean respecto del **inicio** del **main axis** que hayamos definido. Si no le aclaramos el `justify-content` al contenedor, `flex-start` es el **valor por defecto**.



# justify-content: flex-end

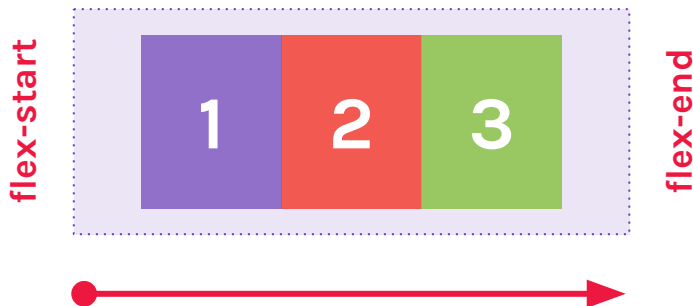
Los ítems se alinean respecto del **final** del **main axis** que hayamos definido.





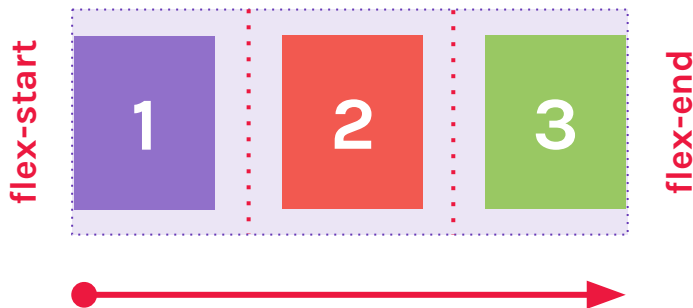
# justify-content: center

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



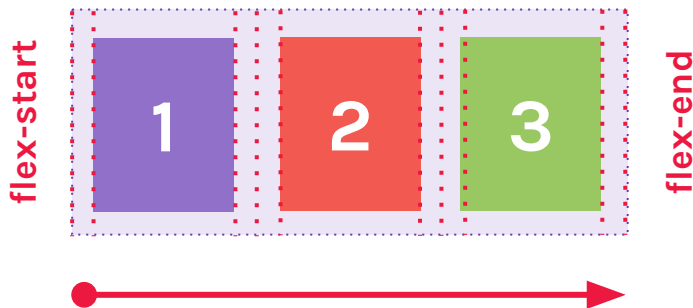
# justify-content: space-between

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

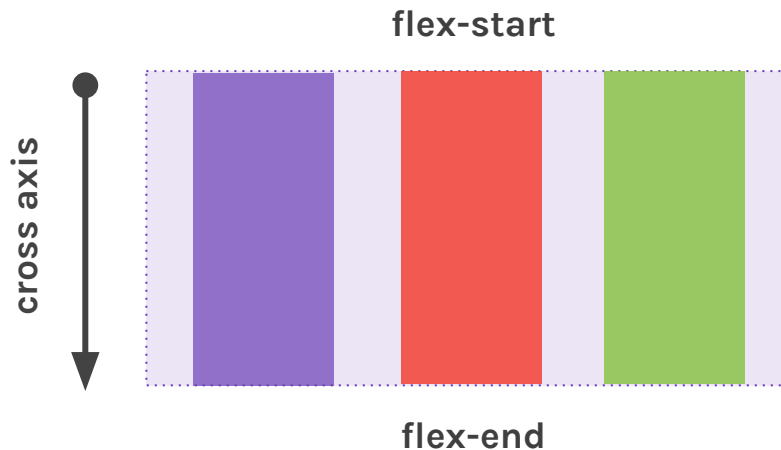
Los ítems se **distribuyen** de manera **uniforme**. El espacio libre disponible se repartirá entre todos los elementos. Del espacio que le toque a cada elemento, la mitad irá a la derecha y la otra a la izquierda (o arriba y abajo en caso de que sean columnas).



# 4 | align-items

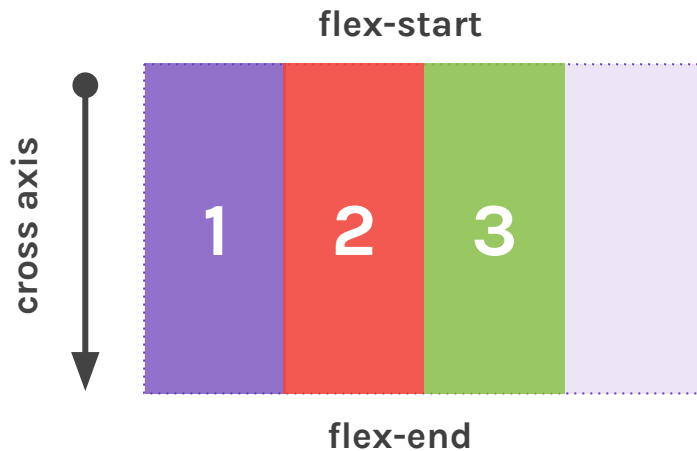
# align-items

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**.



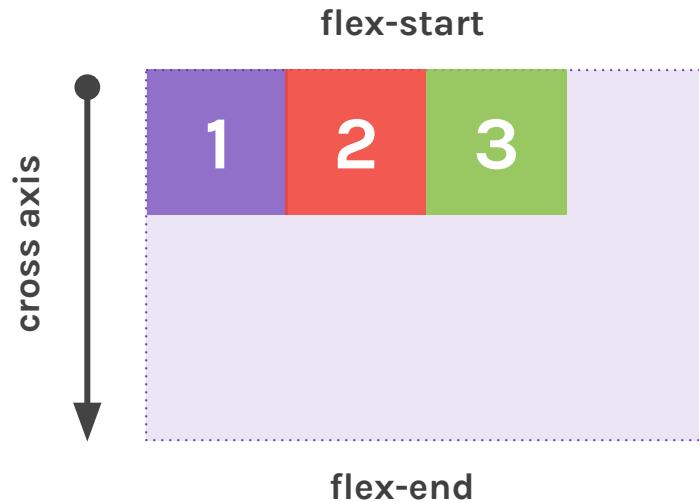
# align-items: stretch

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**.



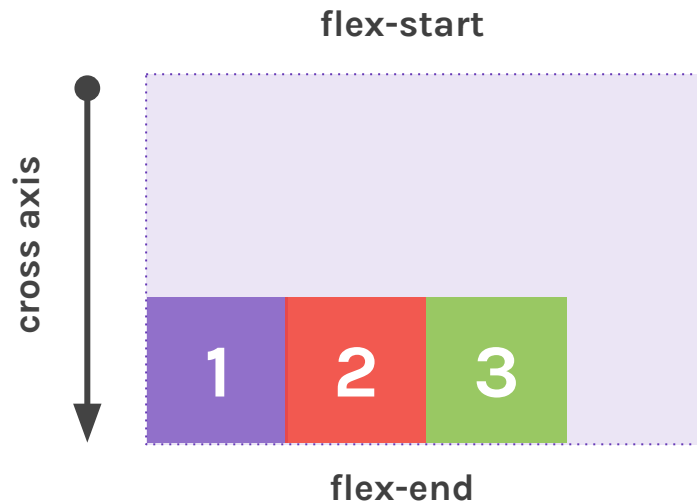
# align-items: **flex-start**

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



# align-items: flex-end

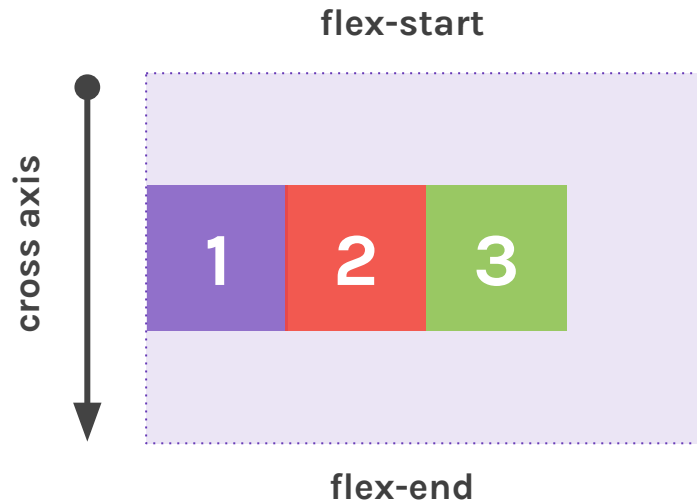
Los **ítems** se **alinean** al **final** del eje transversal.





# align-items: center

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



# 5 | align-content

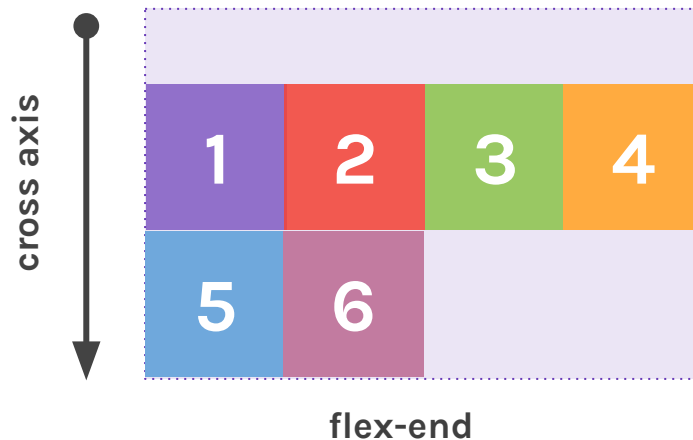
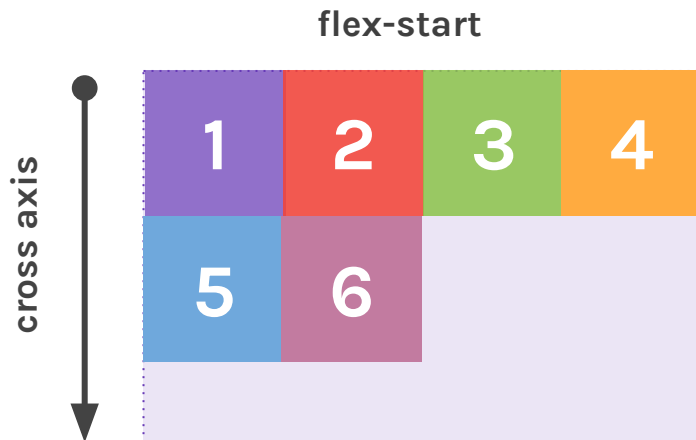
# align-content

Si tenemos un contenedor de una sola línea (donde flex-flow se establece como no-wrap) utilizaremos align-items, pero en el caso de que estemos trabajando con contenedores **multilínea** debemos utilizar **align-content**.

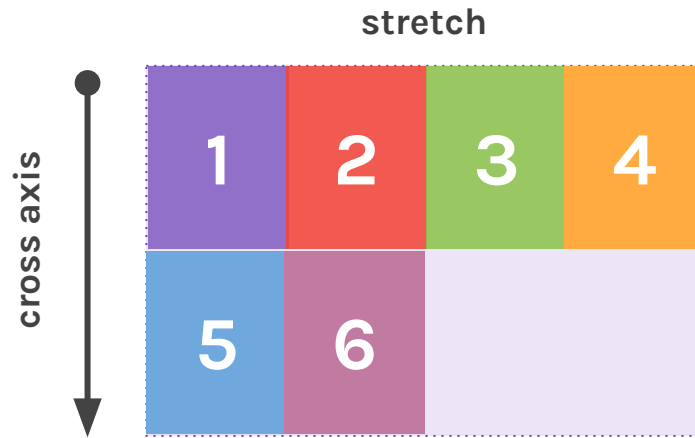
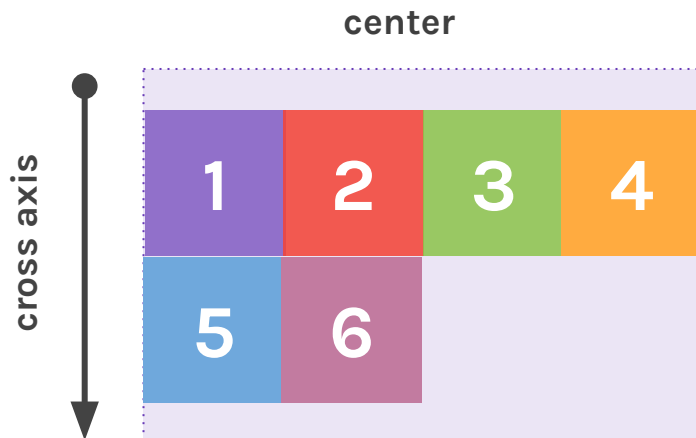
Con esta propiedad **alineamos** los **ítems** a lo largo del **cross axis** cuando los contenedores flexibles incluyen de varias líneas (donde **flex-flow** se establece en **wrap** o **wrap-reverse**).

Los valores que admite la propiedad align-content son similares a los que podemos utilizar para align-items. Vale la pena probar como funcionan ambas propiedades para entenderlas bien.

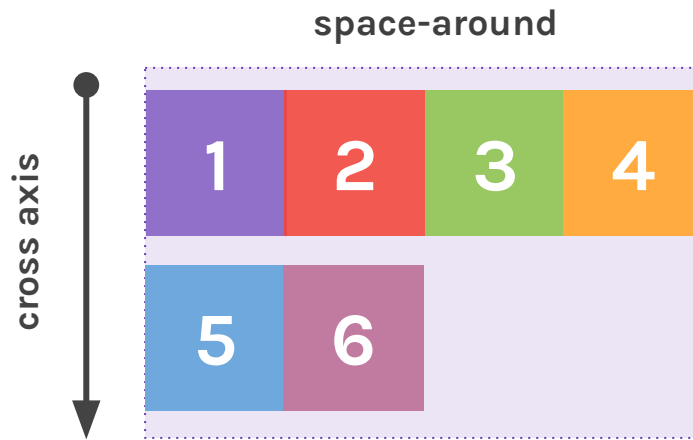
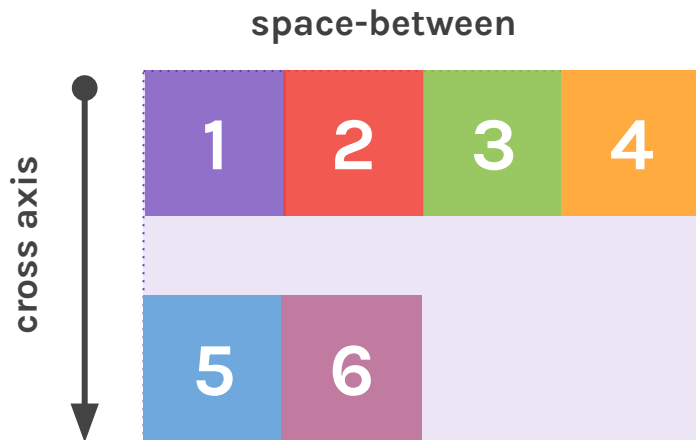
# align-content: **flex-start** | **flex-end**



# align-content: center | stretch



# align-content: **space-between** | **space-around**



DigitalHouse>  
Coding School