

LOS ÍTEMS

“

Flexbox nos da la posibilidad de aplicarle **propiedades** directamente a cada **ítem** para poder **manipularlos** por **separado** y tener más control.



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** como **valor**.

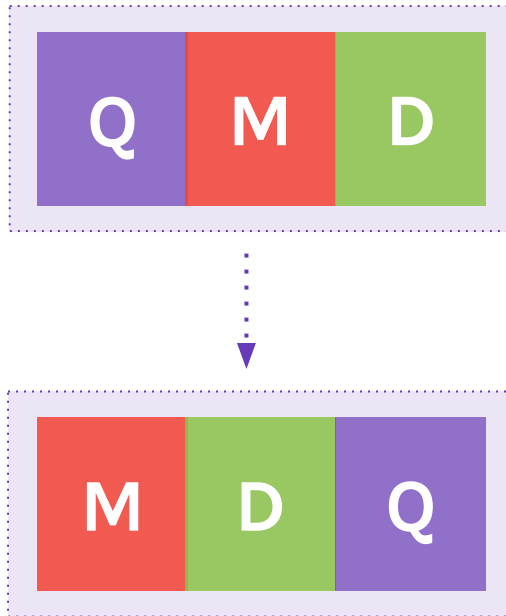


order: número positivo

Si le asignamos a la **caja Q** la propiedad `order` con valor **1**, la misma pasará al final de la fila por ser el número más alto.

Por defecto, el valor del orden de cada ítem es 0.

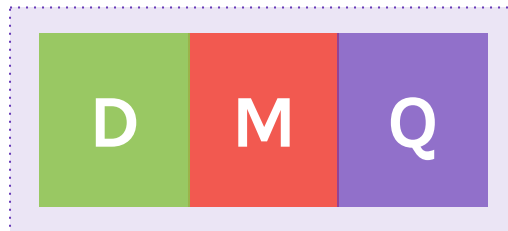
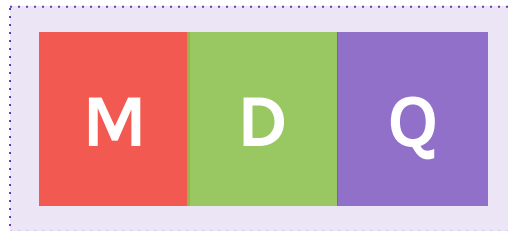
```
.cajaQ {  
    order: 1;  
}
```



order: número negativo

Si ahora le asignamos a la **caja D** la propiedad `order` con un **-1** como valor, la misma pasará al **principio** de la fila.

```
.cajaD {  
    order: -1;  
}
```



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.

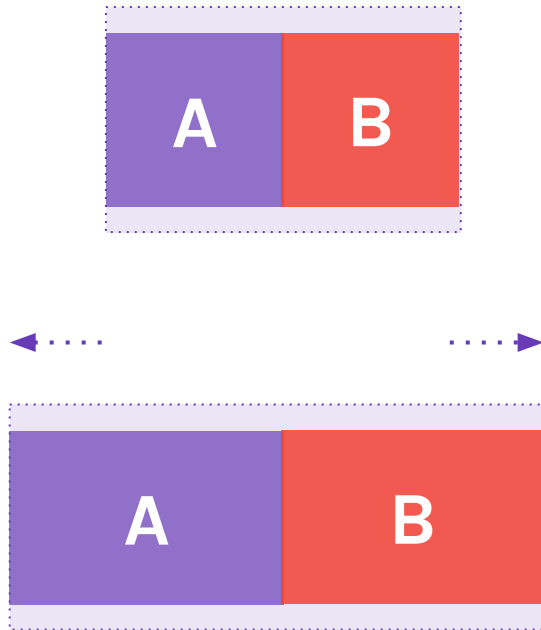
Configura un crecimiento flexible para el elemento.



flex-grow

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.

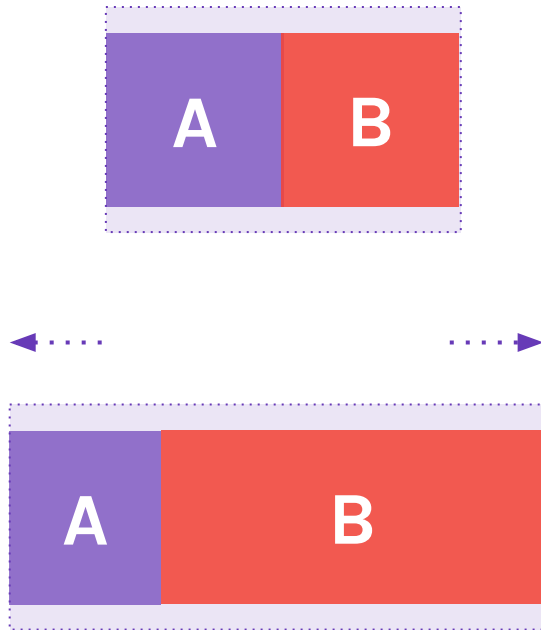
```
.cajaA, .cajaB {  
    flex-grow: 1;  
}
```



flex-grow

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

```
.cajaB {  
    flex-grow: 1;  
}
```



El número que le asignamos a **flex-grow** determina qué cantidad de espacio disponible dentro del contenedor flexible tiene que ocupar ese ítem.



align-self

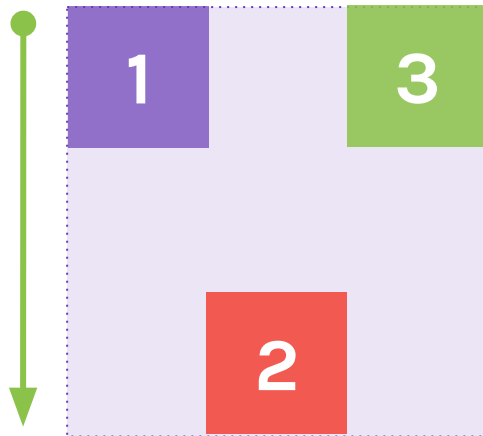
Nos permite **alineal**, sobre el **cross axis**, a cada ítem al que le **apliquemos** esta propiedad, independientemente de la **alineación** que se haya definido en el **contenedor flex** con **align-items**.



align-self: flex-end

Con `flex-end` el **ítem** se alinea al **final** del eje transversal.

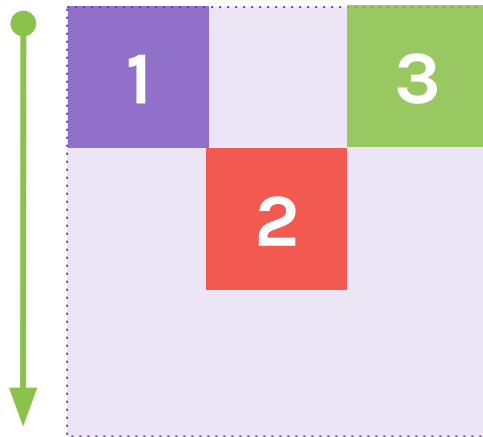
```
.contenedorPadre {  
    align-items: flex-start;  
}  
.  
caja2 {  
    align-self: flex-end;  
}
```



align-self: center

Con `center` el ítem se alinea al **centro** del eje transversal.

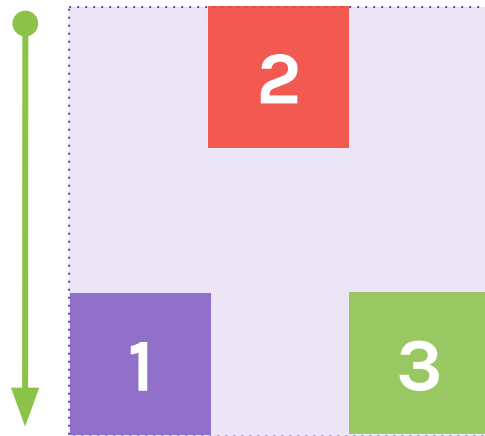
```
.contenedorPadre {  
    align-items: flex-start;  
}  
.  
caja2 {  
    align-self: flex-end;  
}
```



align-self: flex-start

Con `flex-start` el **ítem** se alinea al **inicio** del eje transversal.

```
.contenedorPadre {  
    align-items: flex-end;  
}  
.caja2 {  
    align-self: flex-start;  
}
```

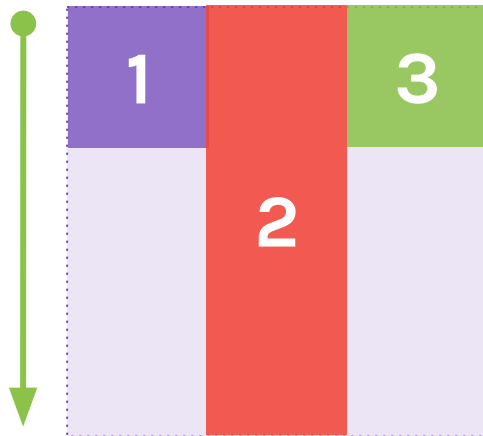


align-self: stretch

Con `stretch`, el ítem se **ajusta** hasta abarcar **todo** el **cross axis**, siempre y cuando no tenga definida una altura y el contenedor padre tenga la regla

`flex-wrap: wrap`

```
.caja2 {  
  align-self: stretch;  
}
```



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

