



# Diseño de Interfaz



# Formato visual

# **Layout de contenedores Display flex**

# Caja flexible

Tiene como meta ser una forma más eficiente de crear diseños, alinear y distribuir espacios entre ítems en un contenedor, incluso cuando las dimensiones de esos ítems son desconocidas y/o dinámicas (en virtud de eso el término “flex”).

Hasta ahora conocemos algunas formas de organizar nuestro contenido a través de diferentes displays: Bloque, inline, inline-block, tablas, combinado con flotaciones o posiciones para ubicar elementos de manera específica fuera del flujo normal.

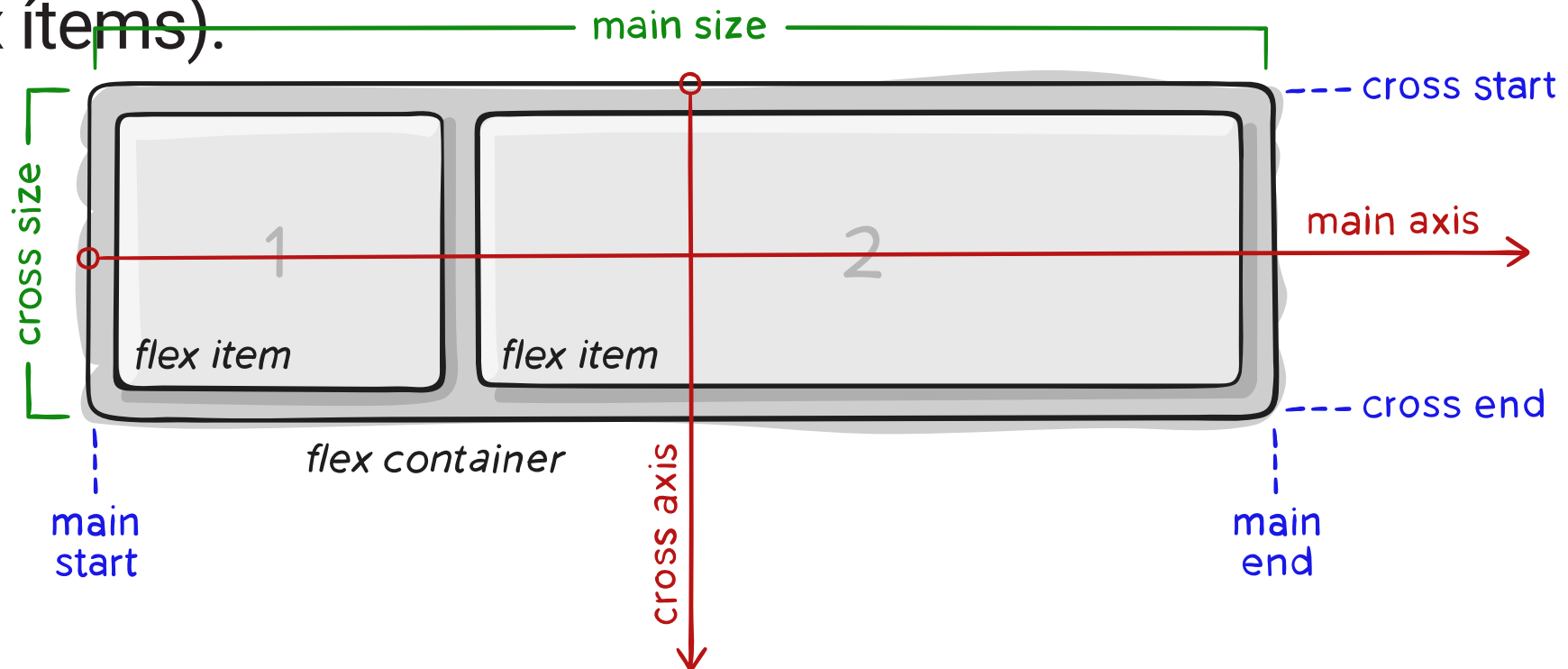
El módulo css de caja flexible (flexbox) es un modelo avanzado de layout que permite manejar con mayor facilidad la ubicación, alineación, tamaños y relación de los elementos dentro de un contenedor.

Trabaja por medio de contenedores e hijos, los cuales se pueden disponer de manera vertical u horizontal, de izquierda a derecha, de arriba a abajo, o viceversa.

Al trabajar con flexbox, vamos a diferenciar contenedor de hijos, cada uno con sus propiedades y características para poder lograr el diseño deseado.

# Elementos

Flex es un módulo completo y no una propiedad única; algunas propiedades deben declararse en el contenedor (el elemento principal, que llamamos de flex container), mientras que otros deben declararse en los elementos secundarios (el flex ítems).



**Los ítems se distribuirán en el diseño siguiendo el eje principal o transversal.**

**Eje principal:** el eje principal de un flex container es el eje primario y a lo largo de él son insertados los flex ítems.

*Precaución: El eje principal no es necesariamente horizontal; Dependerá de la propiedad flex-direction (vea abajo).*

**Main-start | main-end:** los flex ítems se insertan en el contenedor empezando por el lado start, dirigiéndose hacia el lado end.

**Tamaño principal:** El ancho o alto de un flex ítem, dependiendo de la dirección del contenedor, es el tamaño principal del ítem. La propiedad de tamaño principal de un flex ítem puede ser tanto width como height, dependiendo de cuál esté en la dirección principal.

**Eje transversal:** El eje perpendicular al eje principal



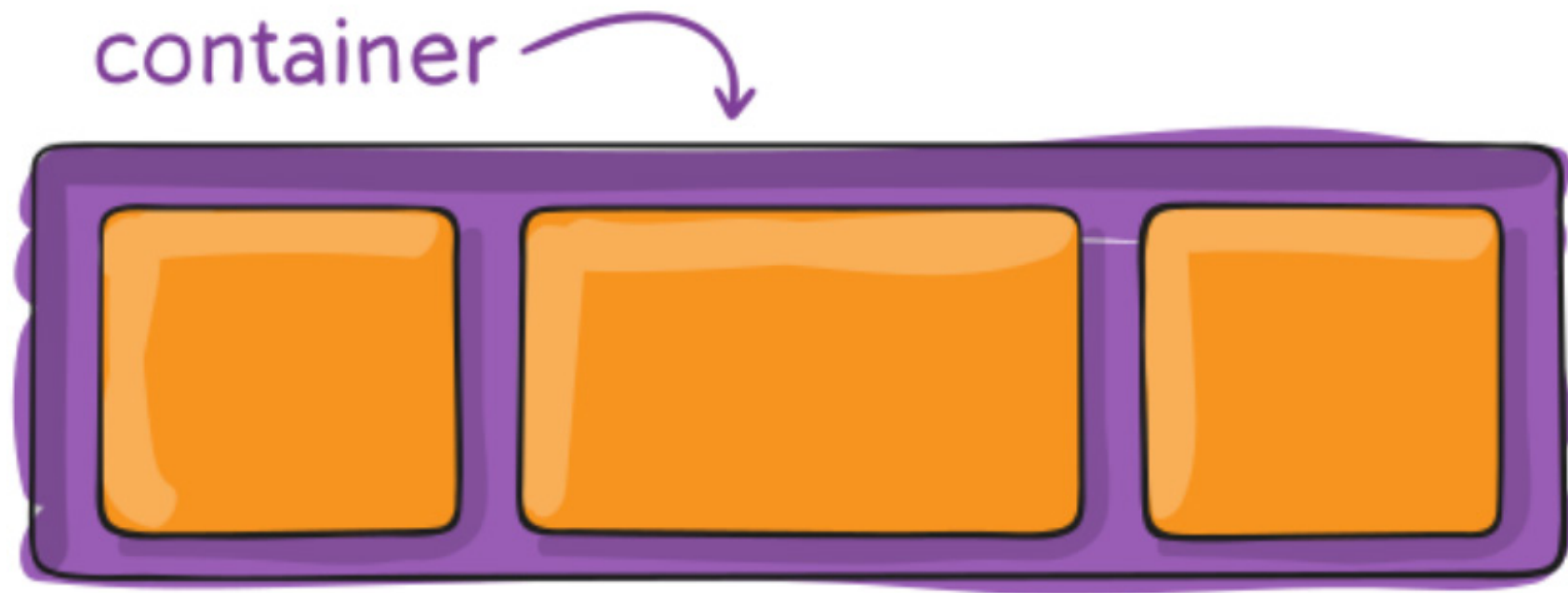
se llama eje transversal. Su dirección depende de la dirección del eje principal.

**Cross-start | cross-end:** Líneas flex se llenan con ítems y se agregan al contenedor, comenzando desde el lado cross start del flex container hacia el lado cross end.

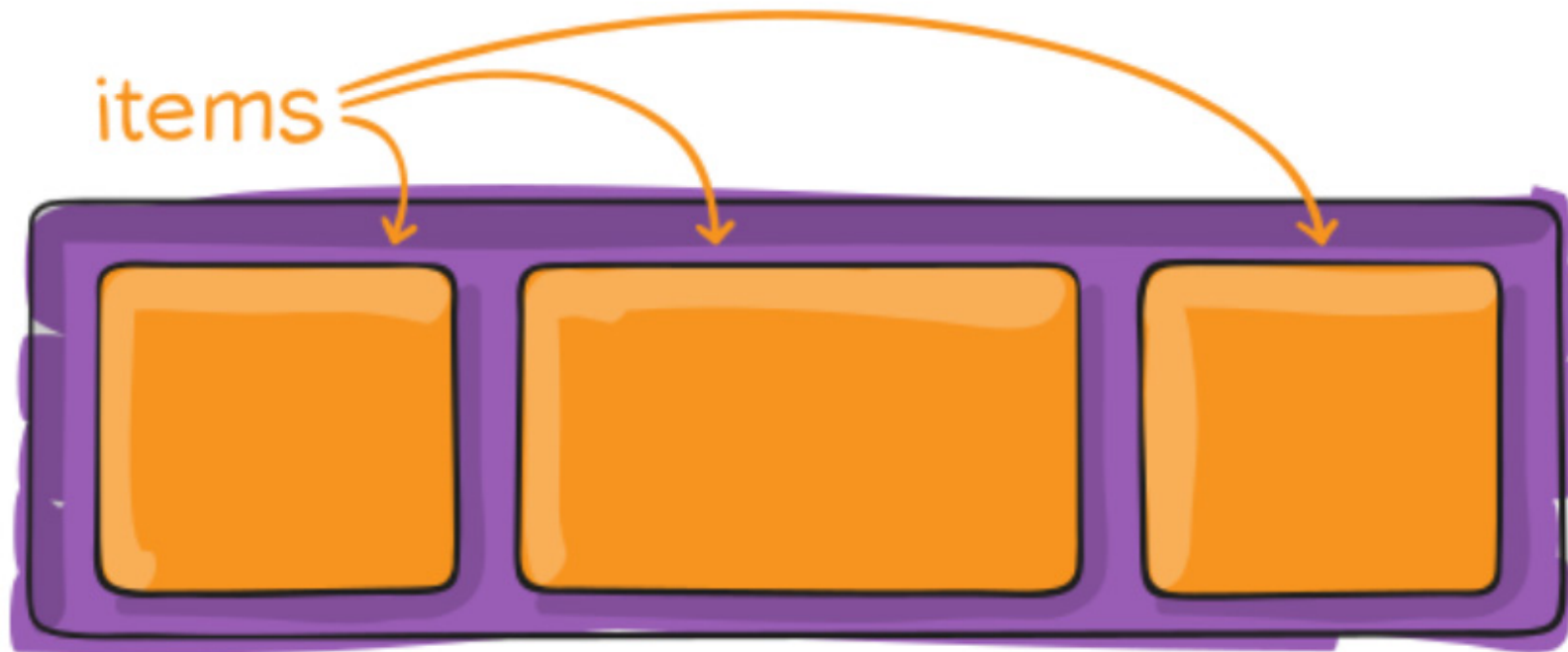
**Cross size:** El ancho o alto de un flex ítem, dependiendo de lo que haya en la dimensión transversal, es el cross size del ítem. La propiedad cross size puede ser el ancho o el alto del ítem, lo que se encuentre en la transversal.

**Flex container** es el elemento que involucra su estructura.

Tu defines que un elemento es un **Flex Container** con la propiedad display y valores **flex** o **inline-flex**.



**Flex Ítem** son elementos secundarios del flex container.



# Flex Container | propiedades CSS

## Display: flex

Esta propiedad define un flex container; inline o block dependiendo de los valores pasados. Coloca todos los elementos secundarios directos en un contexto Flex.

## Flex-direction

Establece el eje principal, definiendo así la dirección en la que los flex ítems están alineados en el flex container.

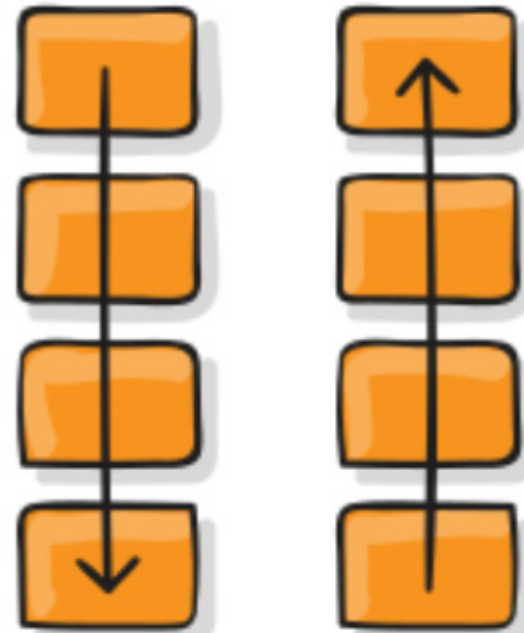
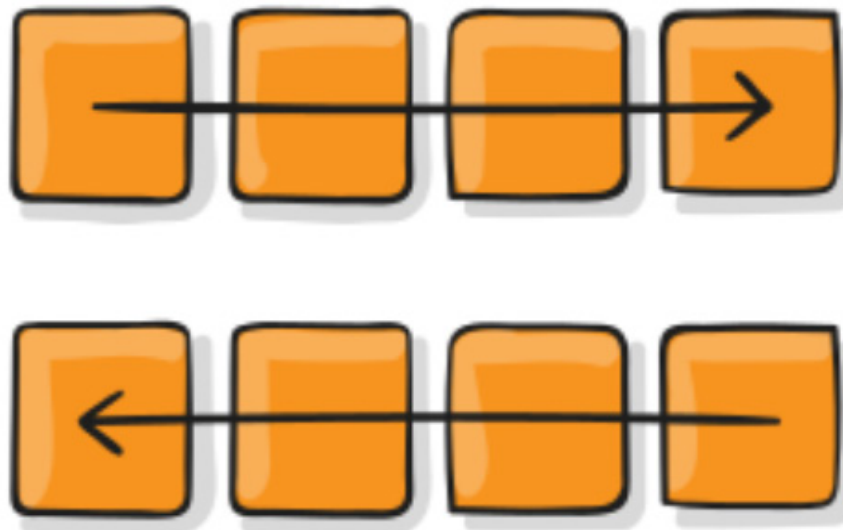
```
flex-direction: row | row-reverse | column |  
column-reverse;
```

**row (estándar):** de la izquierda a la derecha

**row-reverse:** de la derecha a la izquierda

**column:** mismo que row, pero de arriba a abajo

**column-reverse:** mismo que row-reverse pero de abajo hacia arriba.



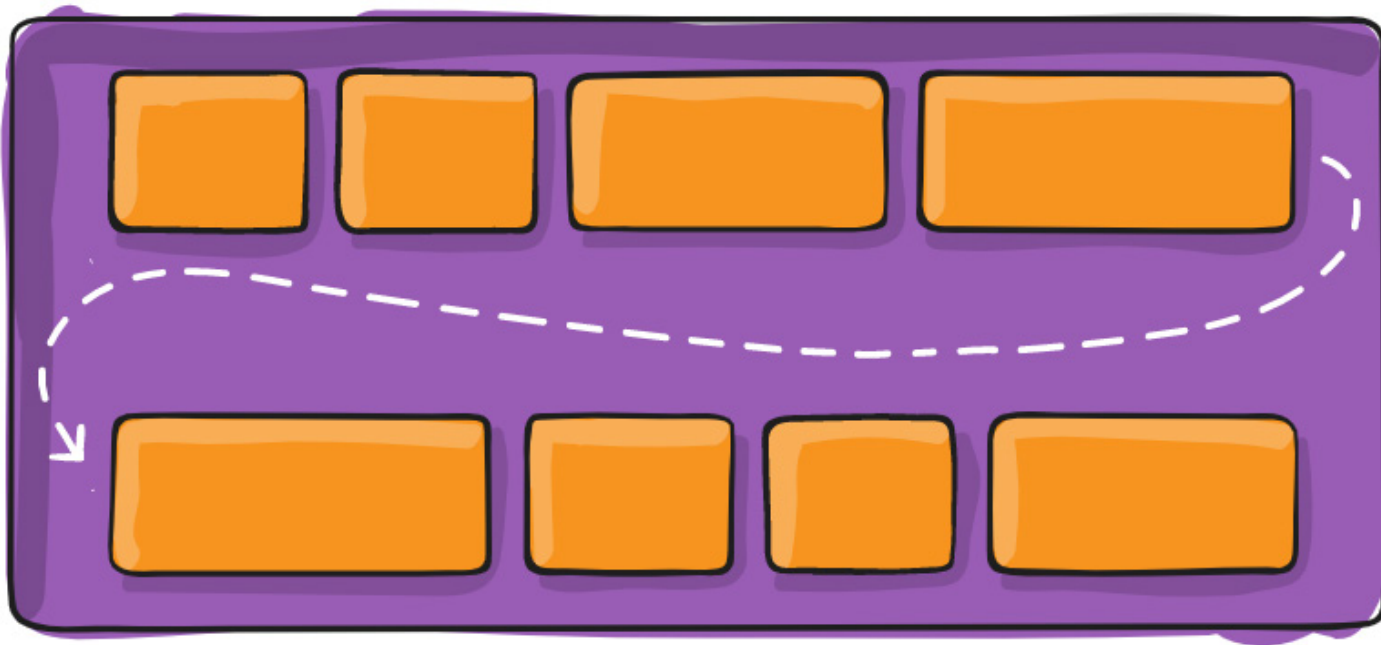
## flex-wrap

Por estándar, los flex ítems todos intentarán encajarse en una sola línea. Con esta propiedad puedes modificar este comportamiento y permitir que los ítems pasen a la siguiente línea según sea necesario.

**flex-wrap: nowrap | wrap | wrap-reverse;**

**nowrap** (estándar): todos los flex ítems estarán en una sola línea

**wrap**: los flex ítems se dividirán en múltiples líneas, de arriba a abajo



## Justify-content

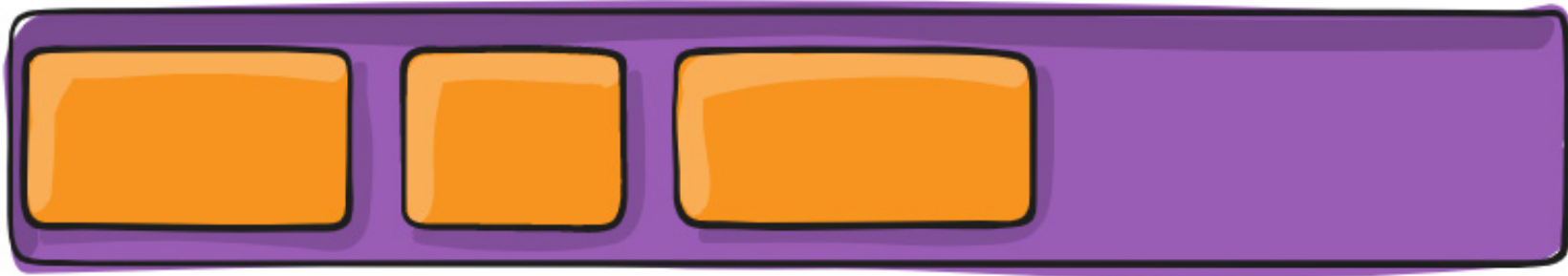
Esta propiedad define la alineación de los ítems a lo largo del eje principal. Ayuda a distribuir el espacio libre que queda en el contenedor, ya sea que todos los flex ítems de una línea sean inflexibles o flexibles, pero ya hayan alcanzado su tamaño máximo. También ejerce cierto control sobre la alineación de los ítems cuando sobrepasan el límite de la línea.

```
justify-content: flex-start | flex-end |  
center | space-between | space-around |  
space-evenly;
```



**flex-start** (estándar): los ítems se alinean a lo largo del borde de inicio (start) de acuerdo con la flex-direction del contenedor.

flex-start



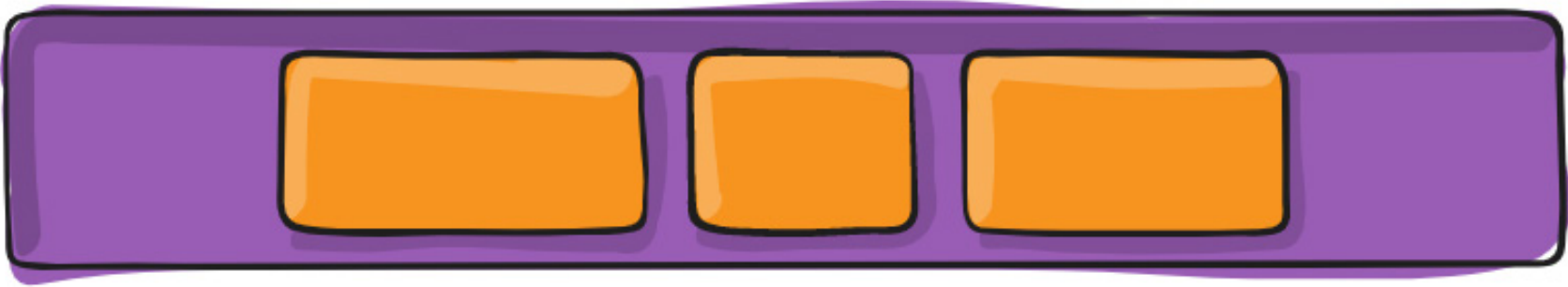
**Flex-end:** los ítems se alinean a lo largo del borde final (end) de acuerdo con la flex-direction del

flex-end



**center:** los ítems están centrados en la línea.

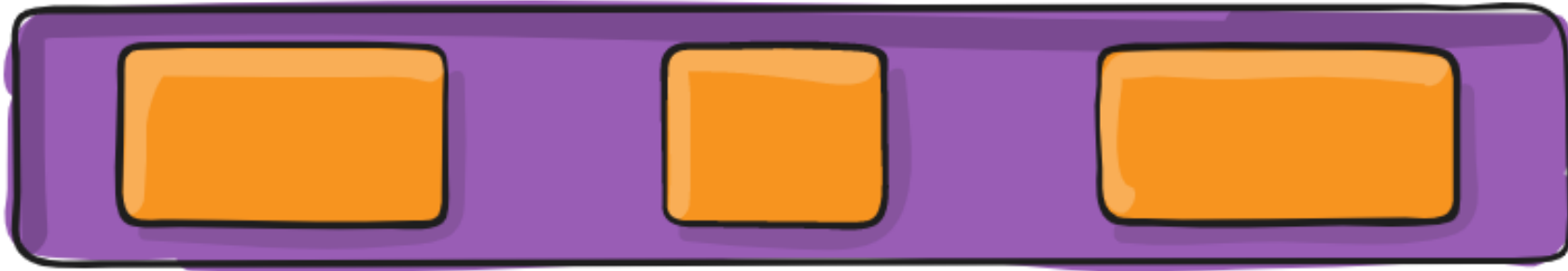
center



**Space-between:** los ítems se distribuyen uniformemente a lo largo de la línea; el primer ítem al lado del borde inicial de la línea, el último ítem al lado del borde final de la línea.

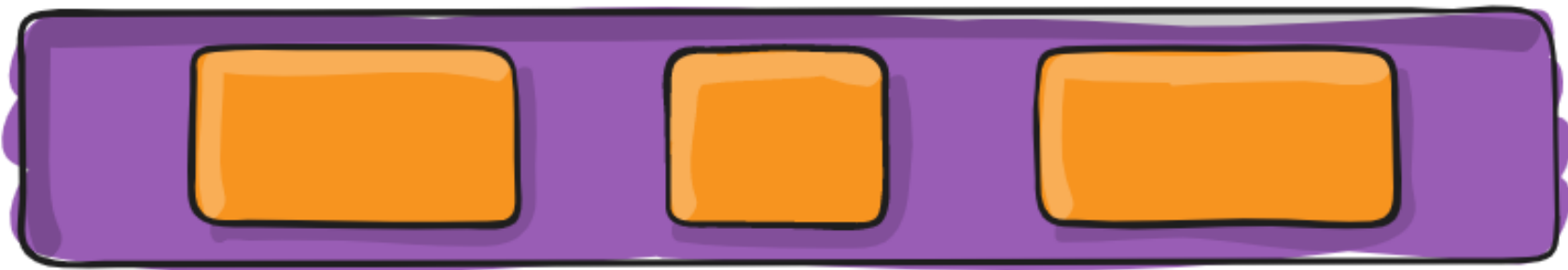
**Space-around:** los ítems se distribuyen en línea con el mismo espacio entre ellos. Tenga en cuenta que visualmente el espacio puede no ser igual, ya que todos los ítems tienen la misma cantidad de espacio en ambos lados: el primer ítem solo tendrá una unidad de espacio a lo largo del borde del contenedor, pero dos unidades de espacio entre él y el siguiente ítem, porque el ítem siguiente también tiene su propio espaciamiento que se está aplicando.

space-around



**Space-evenly:** los ítems se distribuyen de manera que el espacio entre dos elementos cualesquiera en la línea (incluso entre los ítems y los bordes) sea igual.

space-evenly



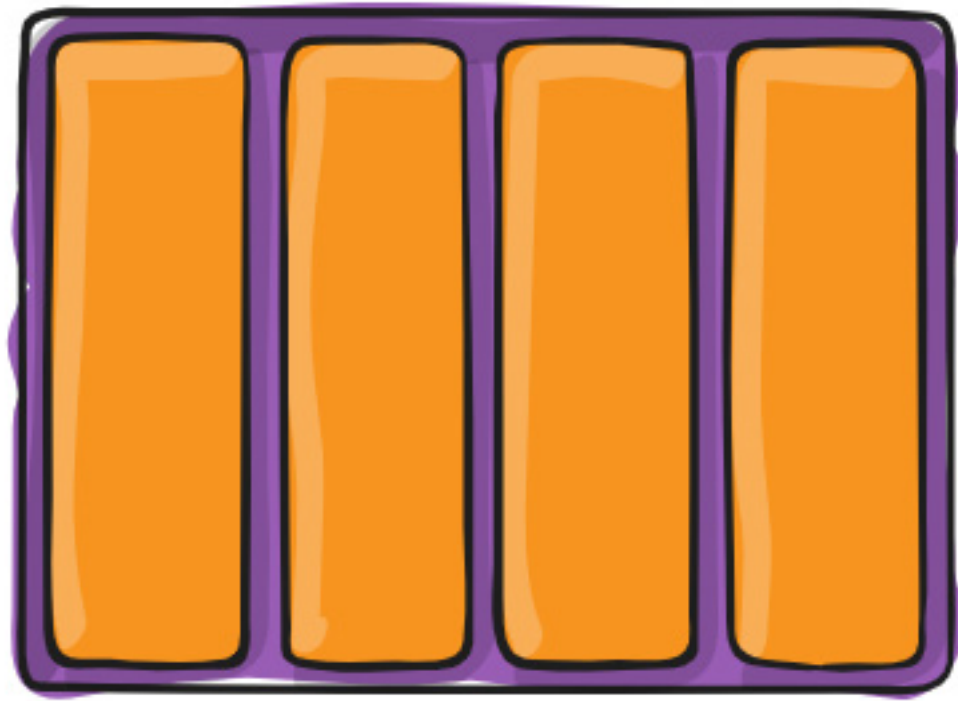
## Align-ítems

Establece el comportamiento estándar de cómo flex ítems están alineados según el eje transversal (cross axis). En cierto modo, funciona de manera similar al justify-content, pero en el eje transversal (perpendicular al eje principal).

```
align-items: stretch | flex-start | flex-end |  
center | baseline;
```

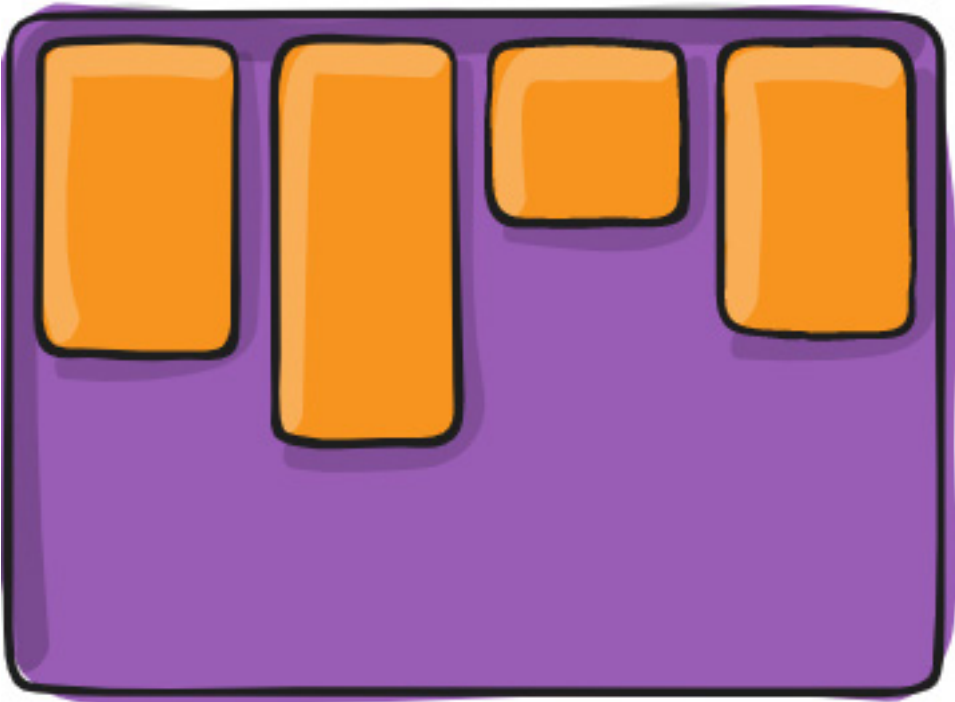
**Stretch (estándar):** estira los ítems para llenar el contenedor, respetando el min-width/max-width).

stretch

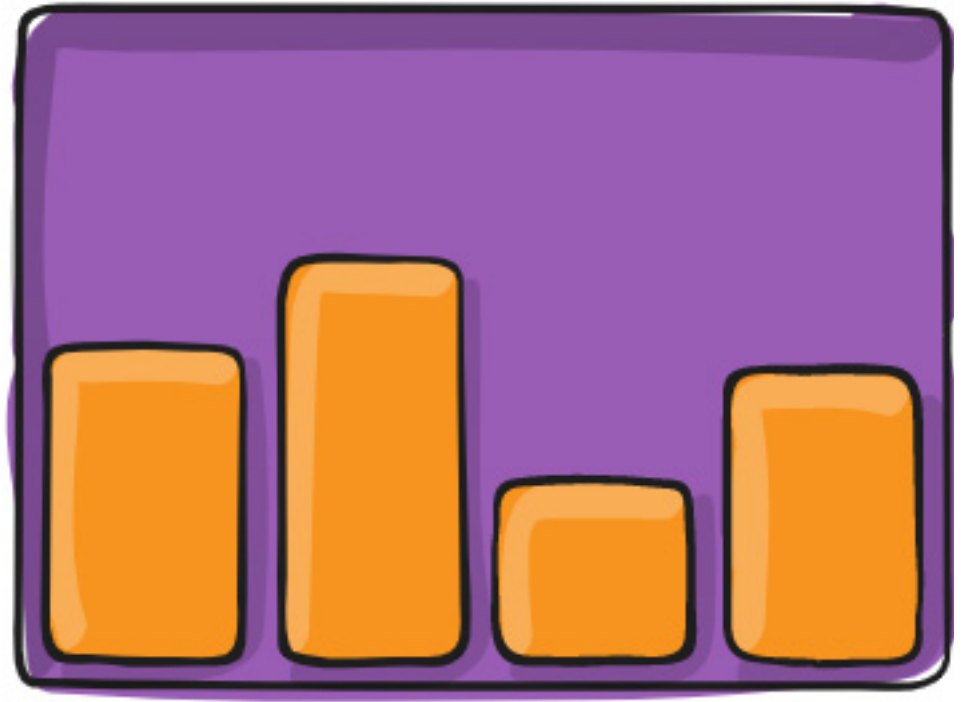


**Flex-start:** ítems se posicionan al inicio del eje transversal.

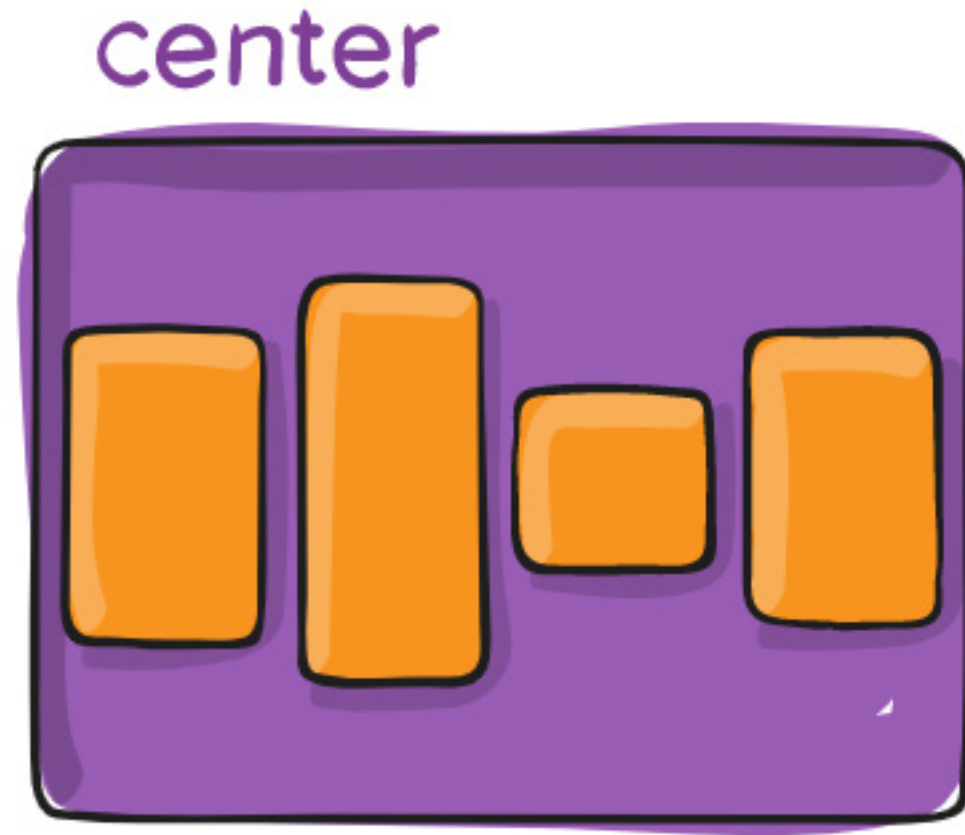
flex-start



flex-end



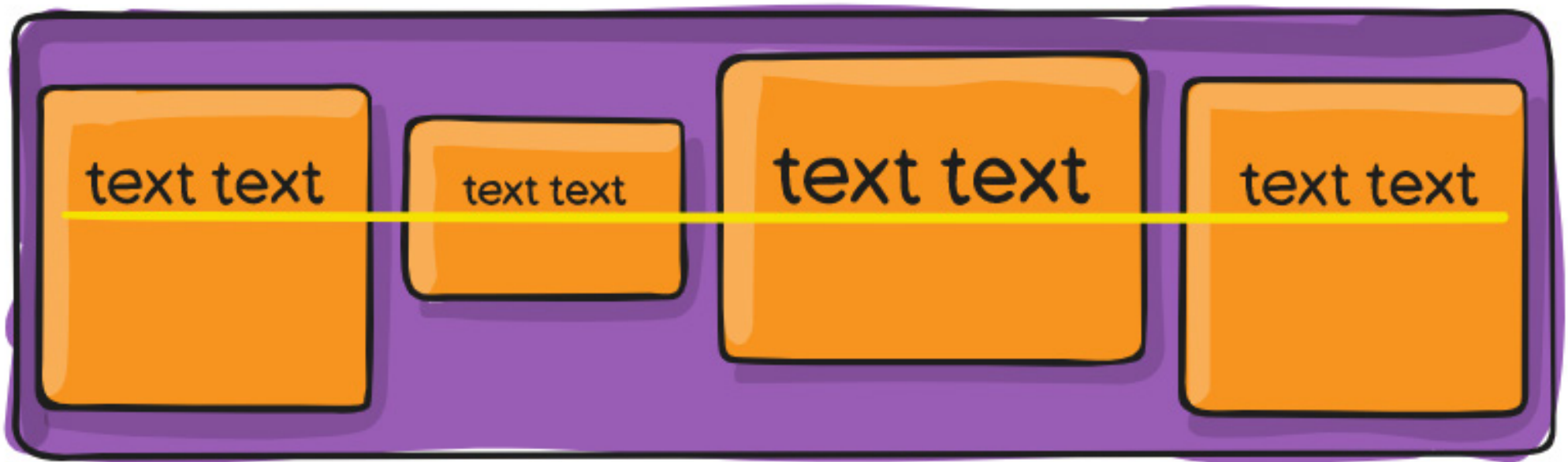
**center:** ítems se centran en el eje transversal.





**baseline:** ítems se alinean de acuerdo con sus baselines.

baseline



## align-content

Organiza las líneas dentro de un flex container cuando hay espacio adicional en el eje transversal, similar a cómo justify-content alinea ítems individuales dentro del eje principal.

*Importante: Esta propiedad no tiene efecto cuando solo hay una línea de flex ítems en el contenedor.*

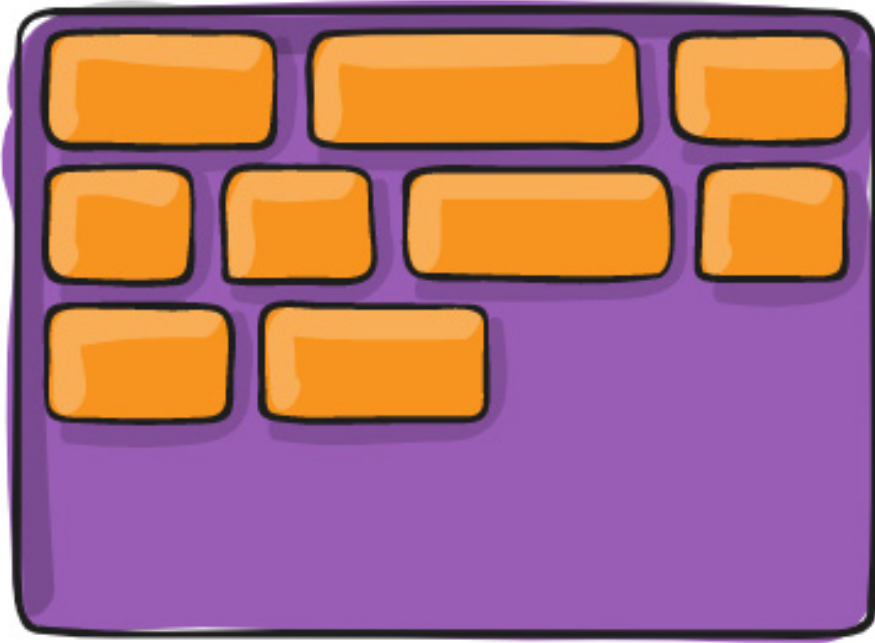
`align-content: flex-start | flex-end | center |  
space-between | space-around | stretch;`

**flex-start:** ítems alineados con el inicio del

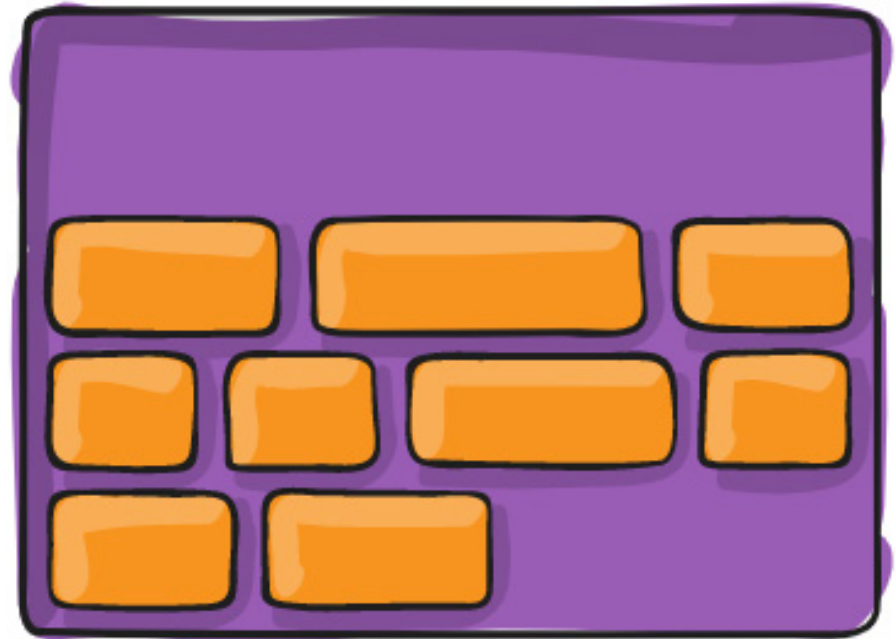
contenedor.

**Flex-end:** ítems alineados con el final del contenedor.

flex-start

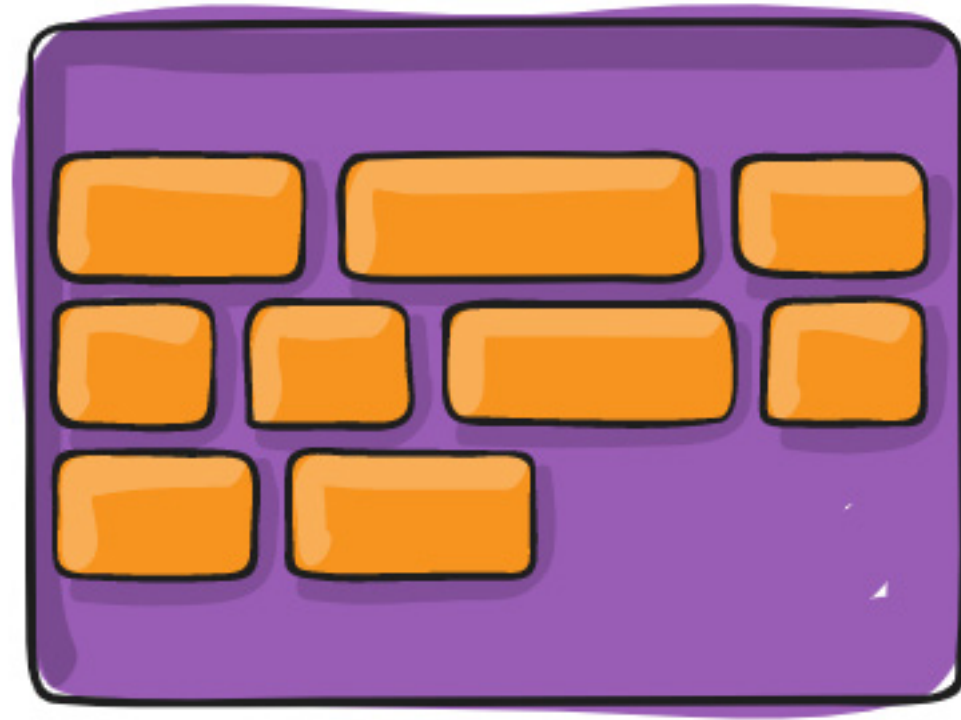


flex-end



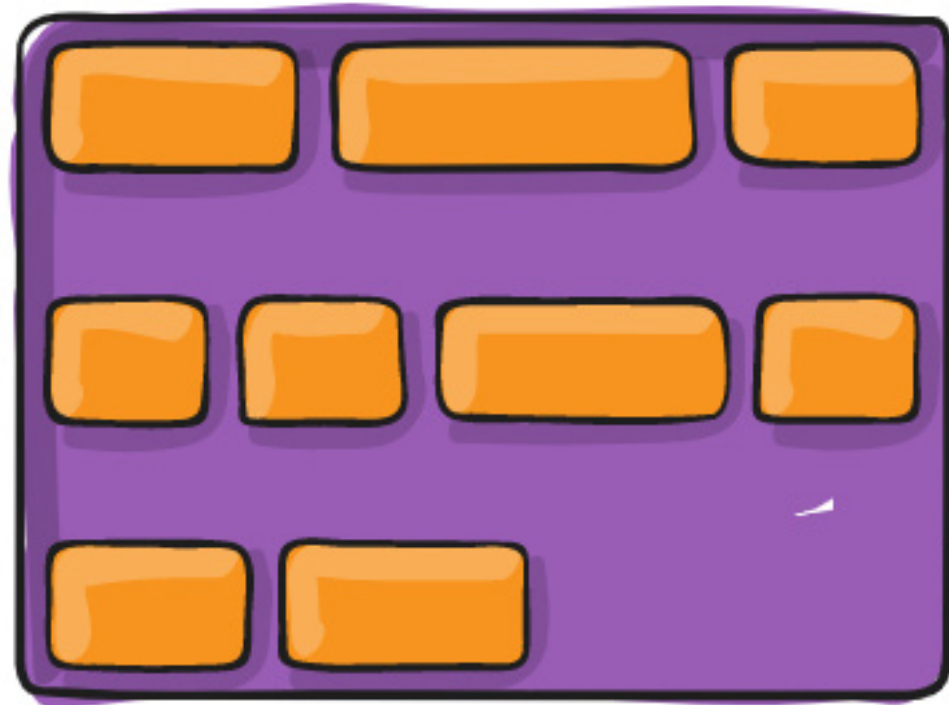
**center:** ítems centrados en el contenedor.

center



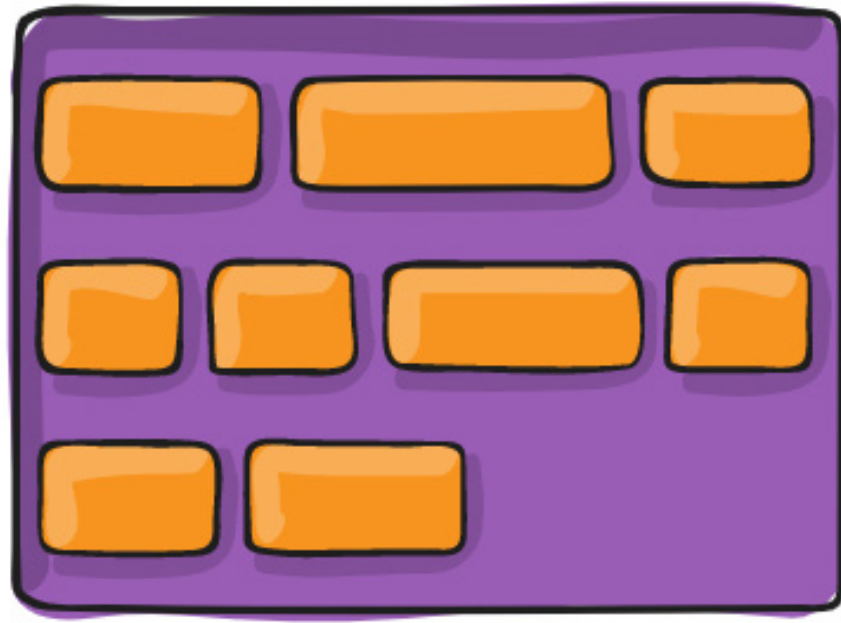
**space-between:** ítems distribuidos uniformemente; la primera línea al inicio del contenedor y la última línea al final del contenedor.

space-between



**space-around:** ítems distribuidos uniformemente con el mismo espaciamiento entre cada línea.

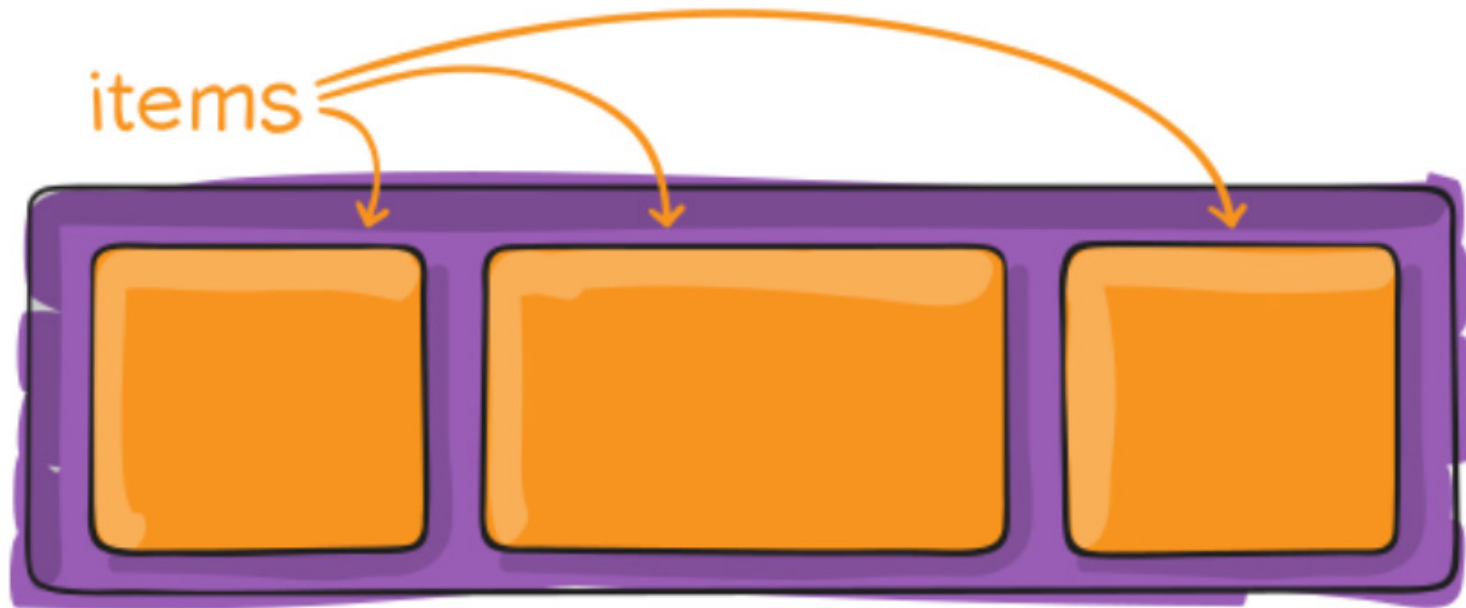
space-around



**space-evenly:** elementos distribuidos uniformemente con el mismo espaciamiento entre ellos.

**stretch (estándar):** ítems en cada línea se estiran para ocupar el espacio remanente entre ellas.

# Flex items | propiedades CSS



Son las propiedades que se pueden modificar de los elementos que son hijos de un elementos flex container



# Definición de tamaño de los elementos

**flex-grow (crecimiento, expansión):** Esta propiedad determina la capacidad de un elemento para expandirse cuando sea necesario. Toma un valor entero que actúa como proporción. La proporción determina cuánto espacio disponible en el contenedor puede ocupar un artículo.

---

Si todos los elementos tienen un crecimiento flexible de 1, el espacio disponible se distribuye uniformemente entre ellos. Si uno de los elementos tiene un crecimiento flexible de 2, ese elemento ocupará el doble del espacio de los otros elementos (o al menos intentará hacerlo).

**flex-shrink (encoger, encoger):** Esta propiedad determina la capacidad de un elemento para comprimir cuando sea necesario.

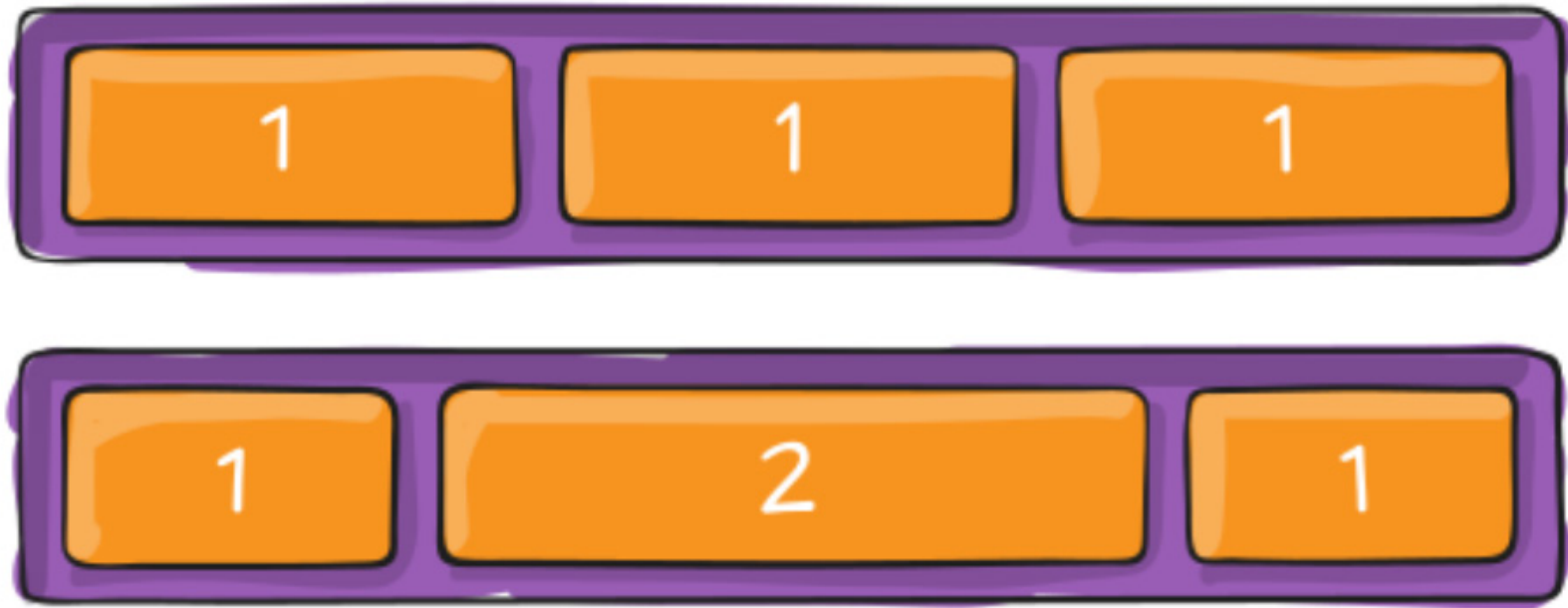
**flex-basis:** Esta propiedad define el tamaño predeterminado del elemento antes de asignar el espacio restante.

Este tamaño puede ser una longitud (por ejemplo, 20%, 5rem, etc.) o una palabra clave.

La palabra clave **auto** significa usar el valor de la propiedad “width” o “height” del elemento.

La palabra clave “**content**” significa tener en cuenta el contenido del elemento.

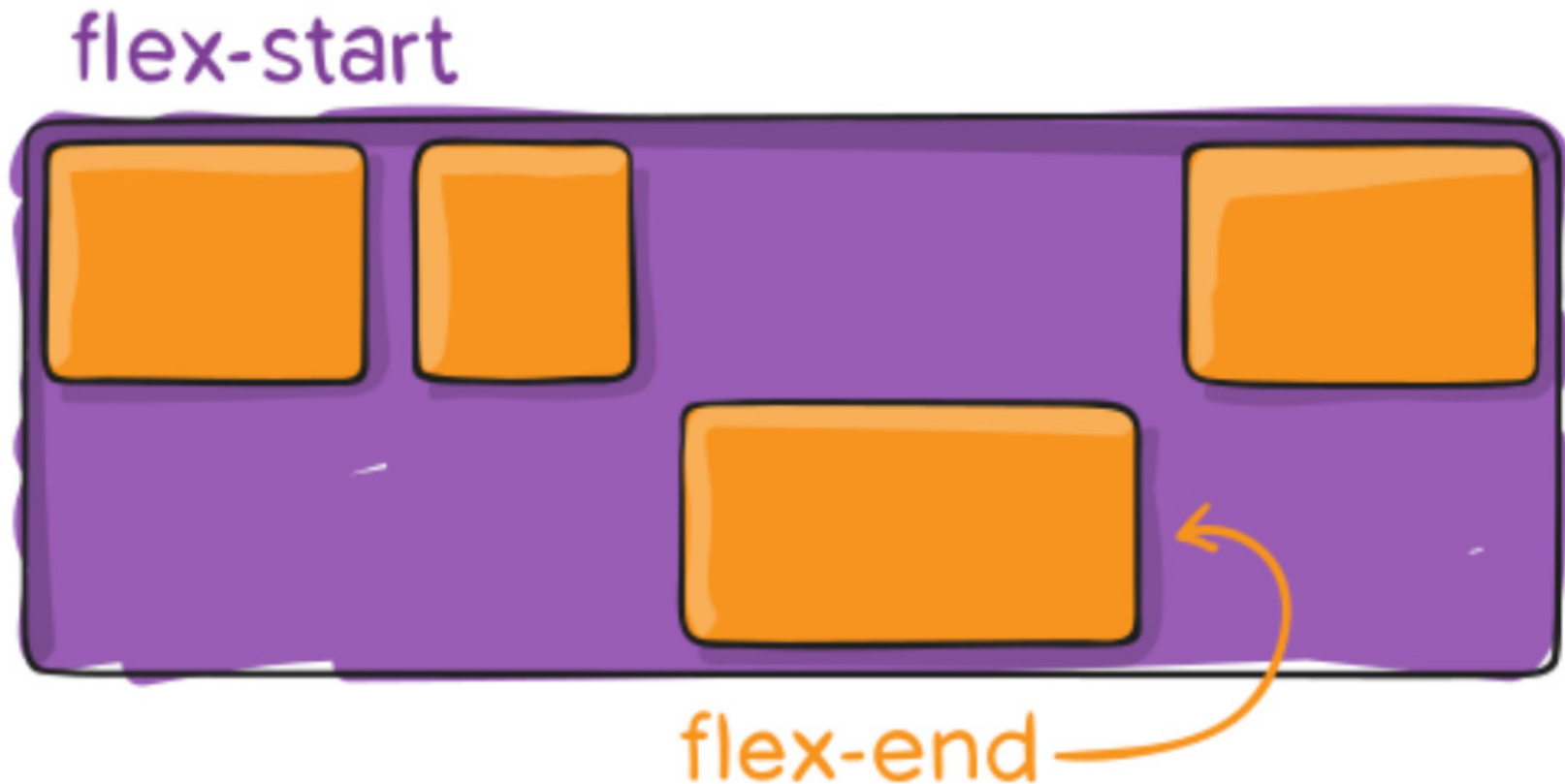
**flex:** Esta propiedad es una abreviatura de flex-grow, flex-shrink y flex-basis. El segundo y tercer parámetro (flex-shrink y flex-basis) son opcionales. El valor predeterminado es 0 1 auto, pero puede omitirse.



```
flex: none | <'flex-grow'> <'flex-shrink'> ||  
<'flex-basis'>
```

## align-self (alineación de un solo elemento):

Esta propiedad le permite sobrescribir la alineación predeterminada o la alineación automática de un elemento individual.



`align-self: auto | flex-start | flex-end | center | baseline | stretch;`