

CSS | Flexbox

Objetivos

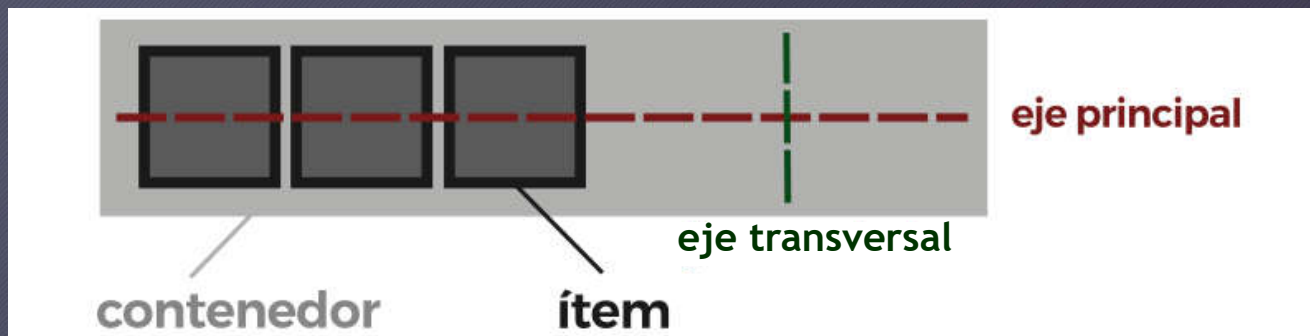
- ✓ Comprender cómo funciona flexbox.
- ✓ Implementar flexbox en sus proyectos.
- ✓ Conocer las diferentes propiedades que podemos aplicar a los contenedores de flexbox y a sus elementos.

Introducción

Flexbox fue diseñado como un modelo unidimensional de layout, y como un método que pueda ayudar a distribuir el espacio entre los ítems de una interfaz y mejorar las capacidades de alineación.

Conceptos

Los elementos básicos de este nuevo esquema:

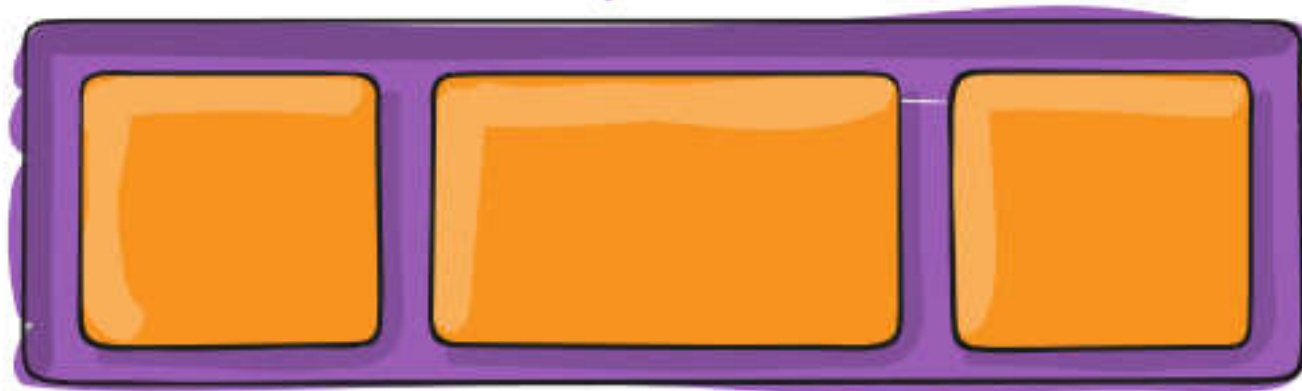


- **Eje principal (main axis):** orientación horizontal (row) - default value
- **Eje transversal (cross axis):** orientación vertical (column)
- **Contenedor (flex container):** elemento padre
- **Ítem (flex ítems):** Cada uno de los hijos flexibles que tendrá el contenedor

Por ejemplo:

```
<div id="contenedor"> <!-- contenedor flex -->
  <div class="item item-1">1</div> <!-- cada ítem flexible -->
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
</div>
```

container



**Properties for the Parent
(flex container)**

Para comenzar:

Utilizamos la propiedad *display* sobre el elemento contenedor y especificamos su valor *flex* o *inline-flex*.

PROPIEDAD	VALOR	DESCRIPCIÓN
Display	inline-flex	Establece un contenedor de ítems flexible en línea
	flex	Establece un contenedor de ítems flexible en bloque

Dirección de los ejes

Existen dos propiedades principales para manipular la dirección y comportamiento de los ítems:

Propiedad	Valor	Descripción
flex-direction:	row row-reverse column column-reverse	Cambia la dirección del eje principal.
flex-wrap:	nowrap wrap wrap-reverse	Evita o permite el desbordamiento (multilinea).

Dirección de los ejes



Veamos un ejemplo

flex-flow

shorthand de las propiedades flex-direction y flex-wrap. El primer valor especificado es flex-direction y el segundo valor es flex-wrap.

```
.box {  
  display: flex;  
  flex-flow: row wrap;  
}
```



Propiedades de alineación de ítems

1. justify-content

2. align-items

3. align-content



1. justify-content

Se utiliza para alinear los ítems del eje principal

```
.container { justify-content: flex-start | flex-end | center | space-between  
| space-around; }
```

← - **justify-content** ----- para eje principal →

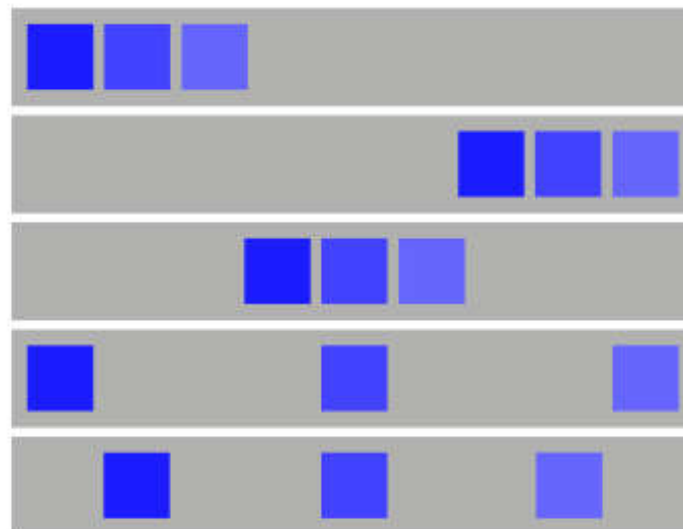
* **flex-start**

flex-end

center

space-between

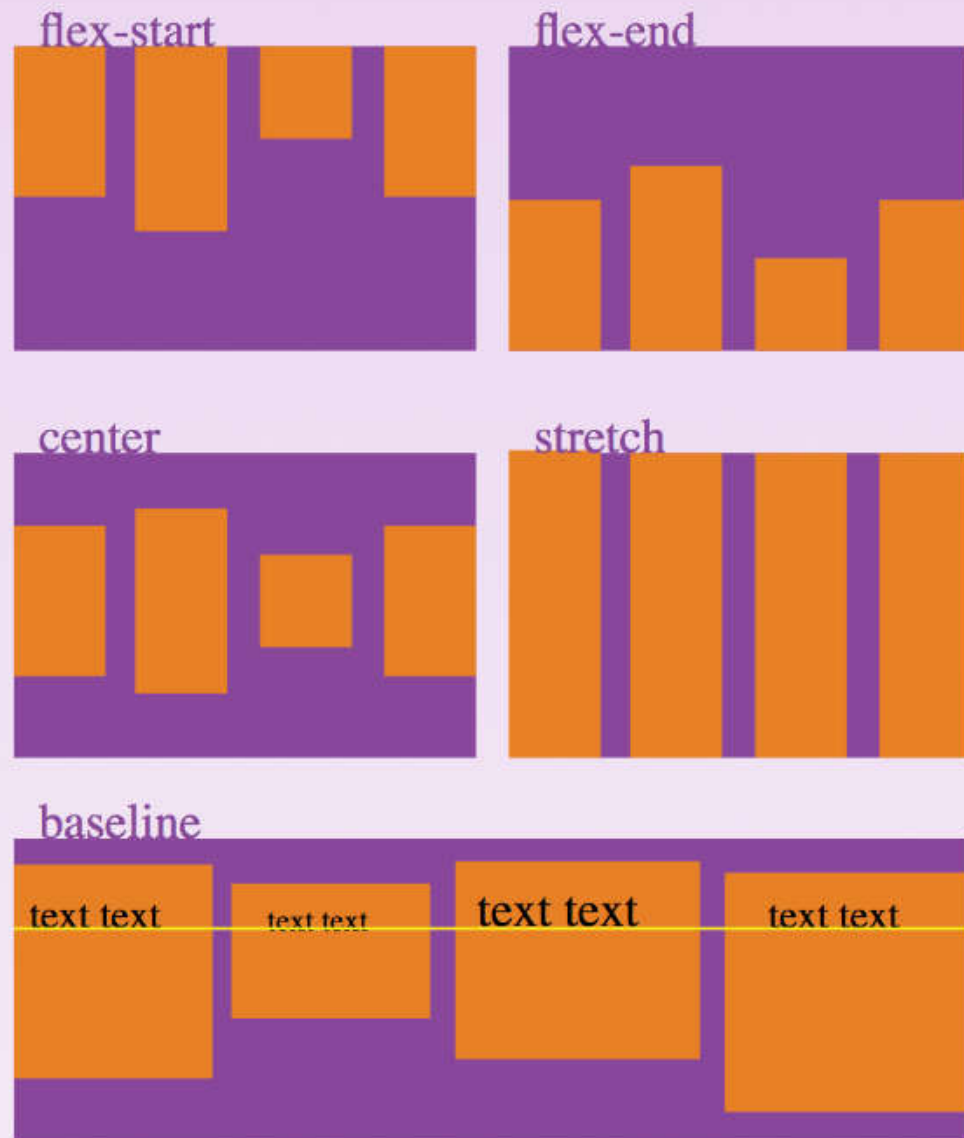
space-around



2. align-items

Alinea verticalmente los ítems flexibles

```
.container { align-items: flex-start | flex-end | center | baseline | stretch; }
```

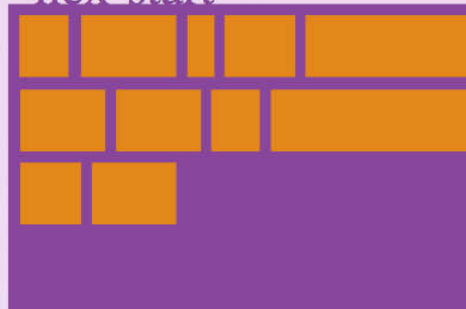


3. align-content

Alinea las líneas de un contenedor flexible cuando hay espacio adicional en el eje transversal (cross axis)

```
.container { align-content: flex-start | flex-end | center | space-between | space-around | stretch; }
```

flex-start



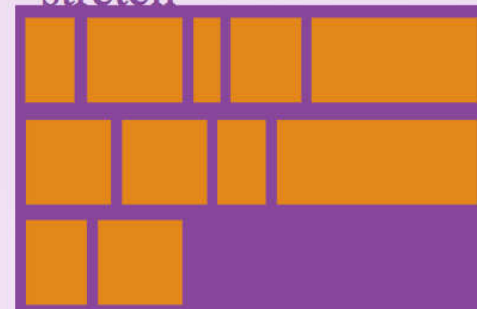
flex-end



center



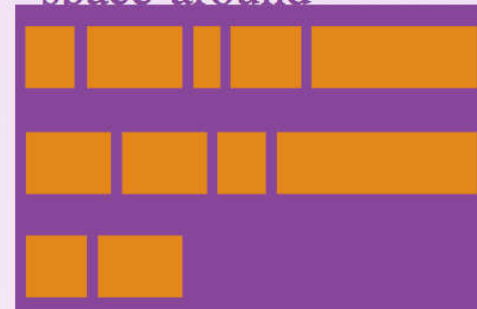
stretch



space-between



space-around



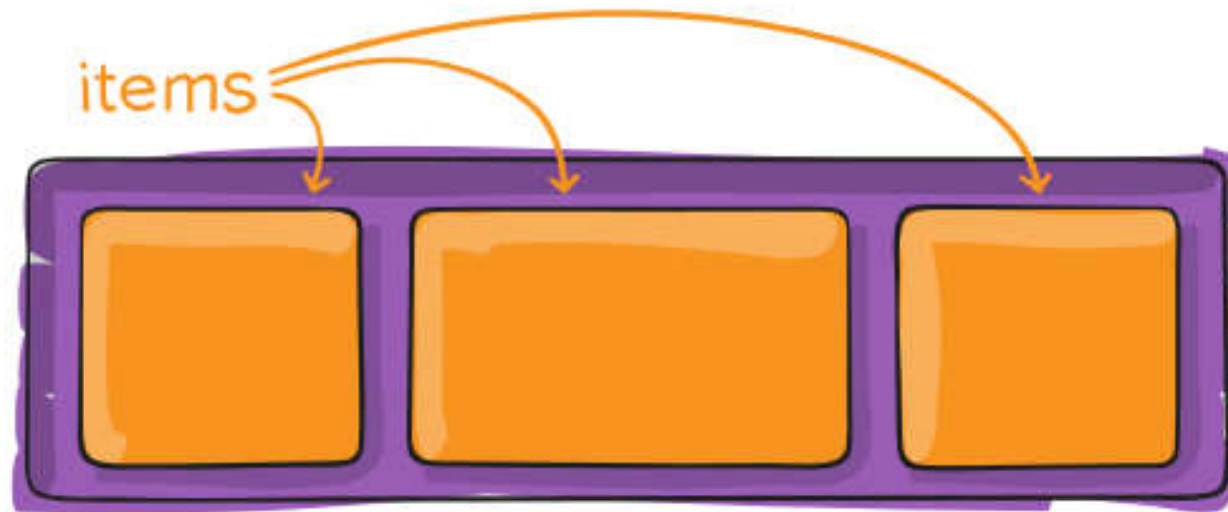
align-ítems y align-content

align-content para establecer cómo múltiples líneas están separadas una de otra. Es decir, align-content determina el espacio entre las líneas, mientras que align-items determina cómo los elementos en su conjunto están alineados dentro del contenedor. Cuando hay solo una línea, align-content no tiene efecto.



Veamos algunos ejemplos aquí:

<https://codepen.io/ironhack/embed/RLZXZR/?height=2743&theme-id=0&default-tab=result&embed-version=2>



**Properties for the Children
(flex items)**

Propiedades de ítems hijos

Todas las propiedades que hemos visto hasta ahora se aplican sobre el contenedor (container flex). Las siguientes propiedades, sin embargo, se aplican sobre los ítems hijos.

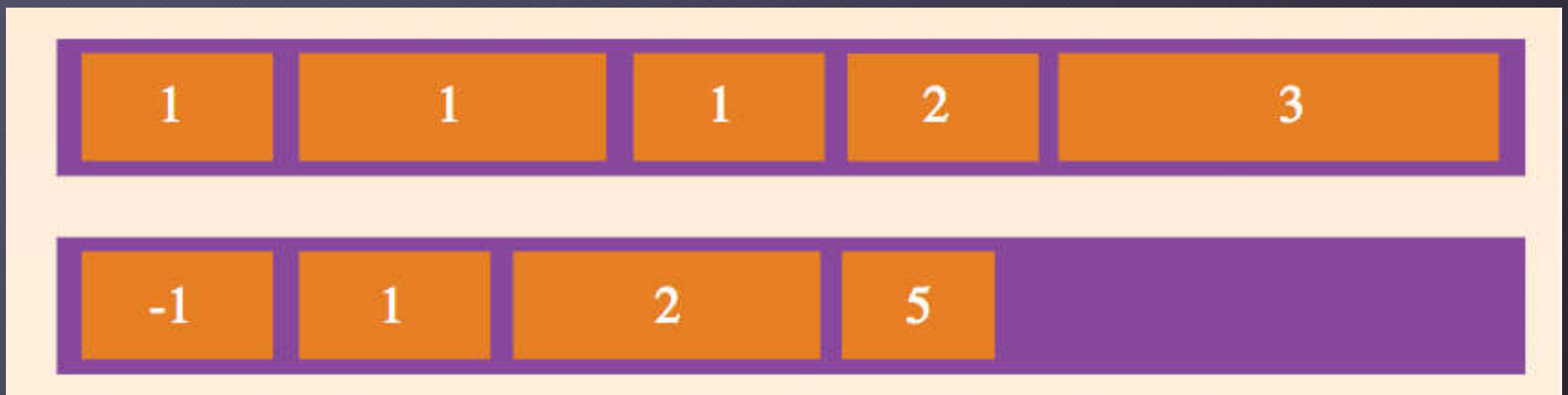
1. Order
2. flex-basis
3. flex-grow
4. flex-shrink
5. flex
6. align-self



1. order

Especifica el orden de un ítem en relación con el resto de los ítems dentro del mismo contenedor.

```
.item { order: <integer>; /* default is 0 */ }
```



Veamos un ejemplo

2. flex-basis

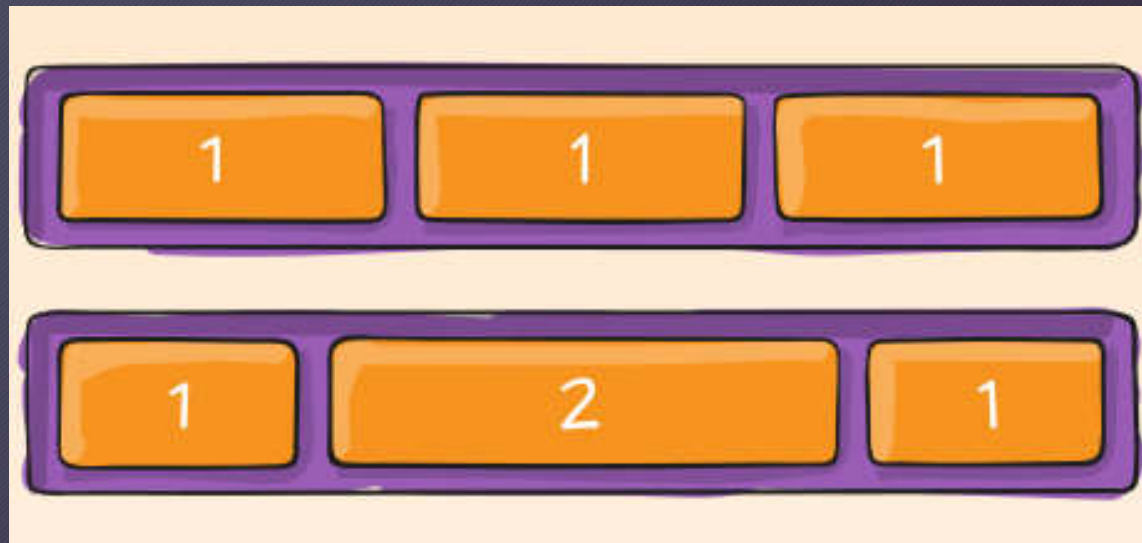
Define el tamaño predeterminado de un elemento antes de que se distribuya el espacio restante. Puede ser una longitud (por ejemplo, 20%, 5rem, etc.) o la keyword *auto*.

```
.item { flex-basis: <length> | auto; /* default auto */ }
```


3.flex-grow

Define la capacidad de un ítem flex de crecer si es necesario. Es decir, define qué cantidad de espacio disponible dentro del contenedor flexible debe ocupar el ítem.

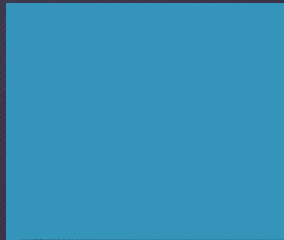
`flex-grow: <number>; /* default 0 */`



4. flex-shrink

Esto define la capacidad de un elemento flexible de encogerse si es necesario.

`flex-shrink: <number>; /* default 1 */`
Los números negativos no son válidos.



5. flex

Shorthand de flex-grow, flex-shrink y flex-basis.

El segundo y tercer parámetro (flex-shrink y flex-basis) son opcionales.

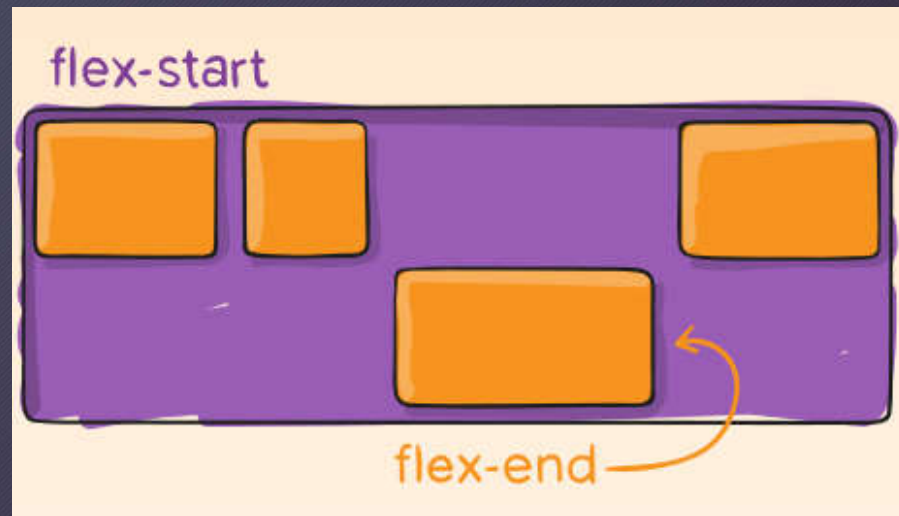
Default value: 0 1 auto



6. align-self

Permite que la alineación predeterminada (o la especificada por align-items) se sobrescriba para items flexibles individuales.

```
.item {  
  align-self: auto | flex-start | flex-end | center | baseline | stretch;  
}
```



float, clear y vertical-align no
tienen efectos en elementos
flexibles

Miremos estos ejemplos:

Te ayudará a comprender mejor todas las propiedades.

[https://codepen.io/ironhack/embed/RLZXZR/?height=2743
&theme-id=0&default-tab=result&embed-version=2](https://codepen.io/ironhack/embed/RLZXZR/?height=2743&theme-id=0&default-tab=result&embed-version=2)

https://www.w3schools.com/css/css3_flexbox.asp

Ejemplo

Imagina que tenemos una navegación típica alineada a la derecha en desktop, pero queremos tener una sola columna en resolución tablet y mobile.

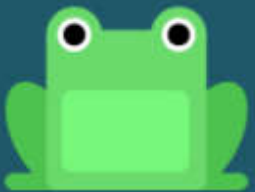
Veamos el ejemplo



Flexbox Froggy

Ayuda a Froggy y a sus amigos escribiendo código CSS en los 24 niveles del juego

<http://flexboxfroggy.com/>



Resumiendo

Flexbox es una herramienta para diseñar los layouts de nuestras páginas web, nos ayuda a alinear y distribuir el espacio entre ítems flexibles en un contenedor flexible.

El uso de flexbox nos asegura que los elementos se comporten de manera predecible y que se adapten a diferentes tamaños de pantalla.

Recursos adicionales

✓ CSS tricks

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

✓ W3schools

https://www.w3schools.com/css/css3_flexbox.asp

✓ MDN

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Conceptos_Basicos_de_Flexbox