



CSS Flexbox

(Repaso)



CSS Flexbox - Repaso

- Podemos utilizar Flexbox colocando **display: flex;** en un contenedor, convirtiéndolo en un **flex-container**.
- Los elementos internos del contenedor pasarán a comportarse como **flex-items** y se posicionarán respecto al **eje principal** de su contenedor.



CSS Flexbox - flex-direction

*flex-direction: **row**;*



*flex-direction: **row-reverse**;*



*flex-direction: **column**;*



*flex-direction: **column-reverse**;*



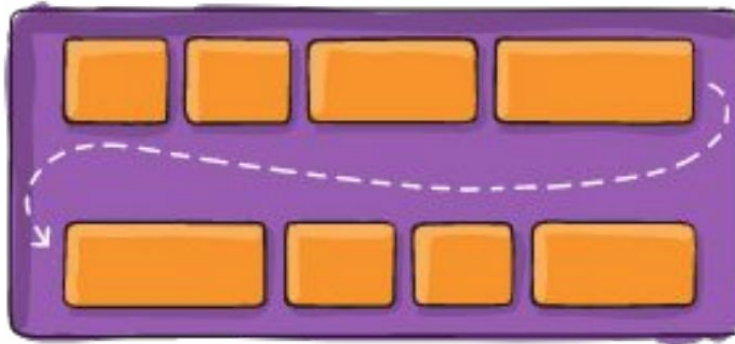
CSS Flexbox - flex-wrap

Por defecto, los **flex-containers** intentan ubicar todos sus elementos en una línea según su eje principal.

Podemos permitirle al contenedor utilizar más de una línea utilizando **flex-wrap: wrap** | **wrap-reverse**;

De esta manera, el contenedor preferirá crear una línea antes de reducir el tamaño de sus elementos internos.

*flex-wrap: **wrap**;*

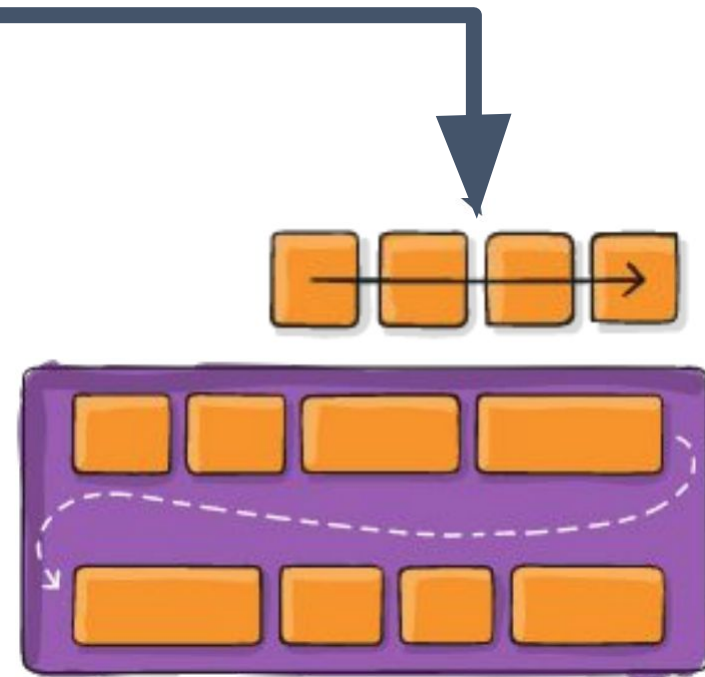


CSS Flexbox - flex-flow

Podemos combinar **flex-direction** y **flex-wrap** en una única propiedad: **flex-flow**.

Ejemplos:

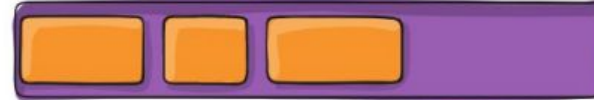
- flex-flow: **row wrap**;
- flex-flow: **column wrap-reverse**;
- flex-flow: **column nowrap**;
- flex-flow: **row-reverse nowrap**;



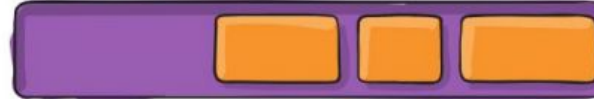
CSS Flexbox - justify-content

Con **justify-content** podemos ubicar los elementos de acuerdo al **eje principal**.

flex-start



flex-end



center



space-between



space-around



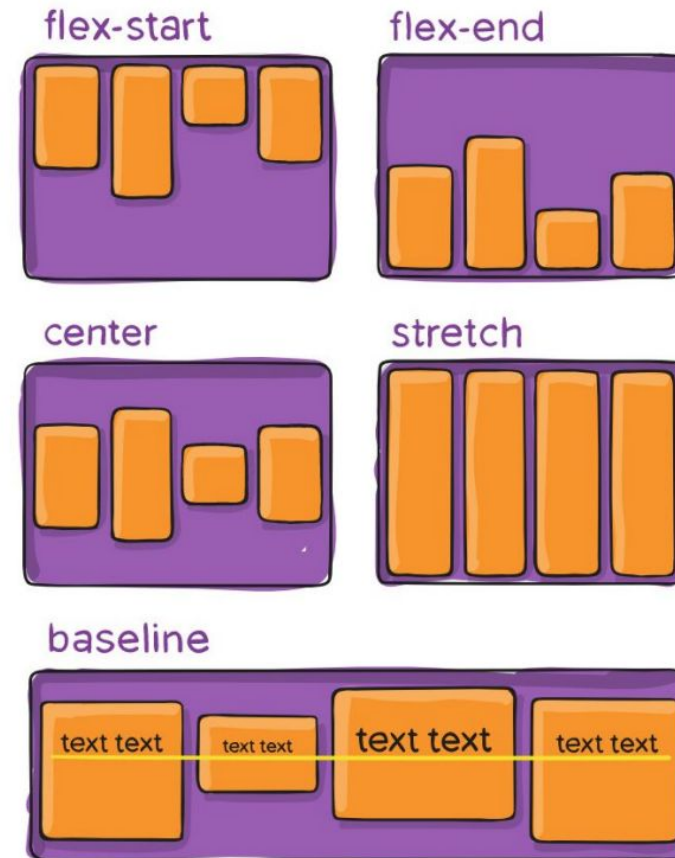
space-evenly



CSS Flexbox - align-items

Con align-items podemos ubicar los elementos de acuerdo al **eje secundario** en la línea actual.

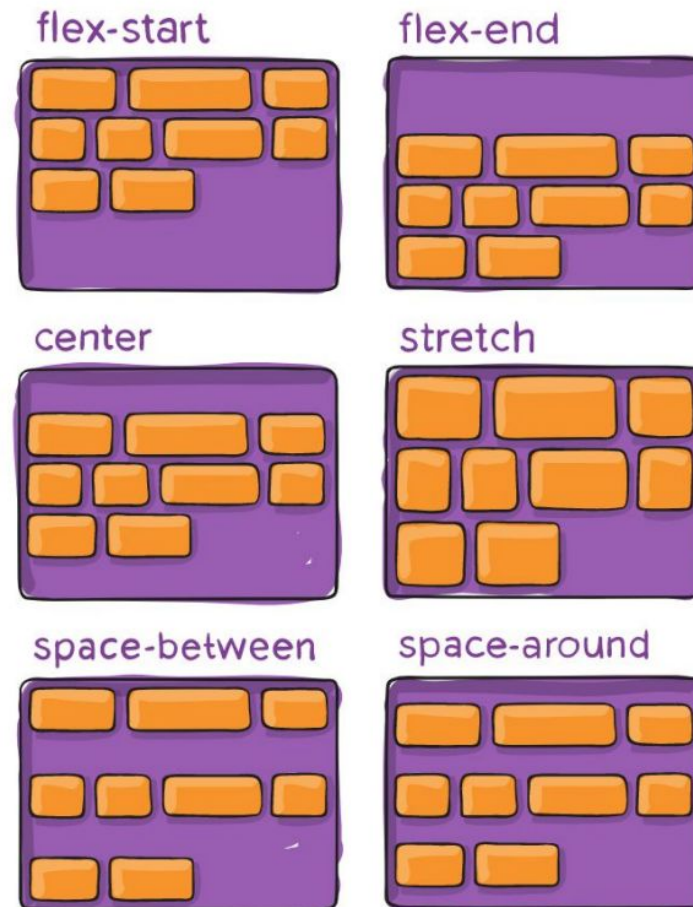
Si el contenedor tiene más de una línea (es decir, **flex-wrap: wrap | wrap-reverse**), la alineación será sobre la línea y no el sobre el total del contenedor.



CSS Flexbox - align-content

Con **align-content** podemos ubicar los elementos de acuerdo al eje secundario sobre el total del contenedor.

Es imposible utilizar **align-content** en un contenedor con una única línea.



CSS Flexbox

(Parte 2)

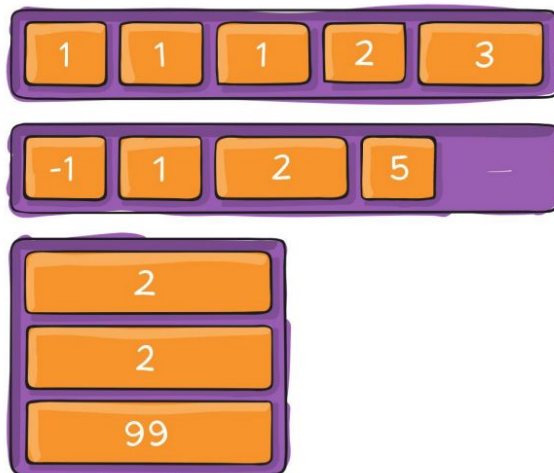


CSS Flexbox - Propiedades de elementos (order)

Una propiedad interesante de los ítems es la de **order**, cuyo valor puede ser cualquier número entero.

El valor por defecto es **order: 0**; por lo que si especificamos **order: -1**; sin tocar el resto de los elementos, éste se colocará al principio de la lista.

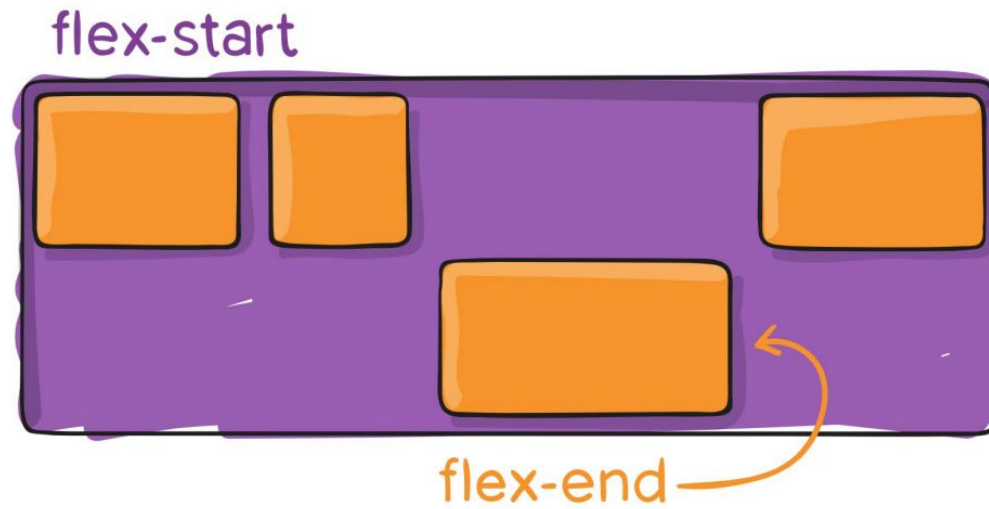
Cambiar el valor de **order** de un elemento sólo lo mueve visualmente, pero su lugar en el HTML es el mismo (cuidado accesibilidad)



CSS Flexbox - Propiedades de elementos (align-self)

La propiedad **align-self** nos permite alinear un elemento en particular de la misma manera que lo haría **align-items**, pero apuntando a un elemento

(...o varios! pero la propiedad debe declararse en **elementos** y no **contenedores**)



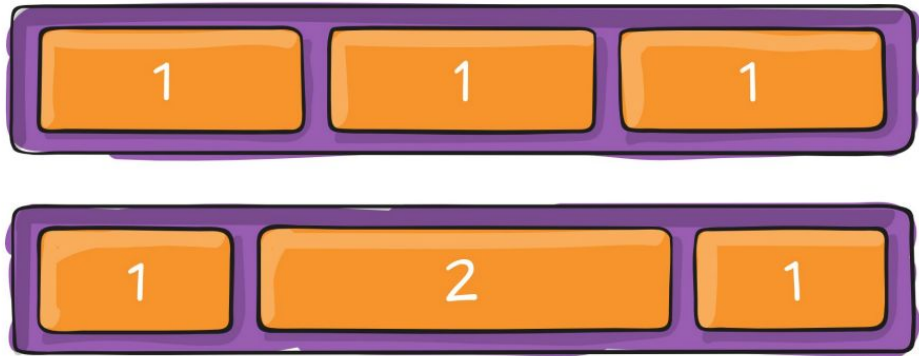
CSS Flexbox - Propiedades de elementos (flex-grow)

La propiedad **flex-grow** define la habilidad de un ítem para “crecer” y tomar el espacio sobrante en el eje principal.

El valor debe ser un número positivo sin unidad (0 incluido) y especifica la proporción del espacio sobrante a tomar.

Por ejemplo, si todos los elementos tienen **flex-grow: 1**; el espacio sobrante del contenedor será distribuido en partes iguales para sus elementos.

Si uno de los elementos tuviese el valor 2, entonces ocuparía el doble de espacio libre que sus hermanos (o intentará).



La propiedad **flex-shrink** define la habilidad de un ítem para “contraerse” con respecto al eje principal.

El valor debe ser un número positivo sin unidad (0 incluido) y especifica una proporción de contracción.

Por ejemplo, si un elemento tiene **flex-shrink: 2;** y otro **flex-shrink: 1;**, el primer elemento se contraerá el doble de rápido que el segundo.

La propiedad **flex-basis** define el tamaño por defecto de un elemento antes de distribuir el resto del espacio.

Puede ser una unidad (**20%**, **5rem**, **230px..**) o una palabra clave (solo Firefox por ahora).

Las palabras clave son las siguientes:

- flex-basis: **fill**;
- flex-basis: **max-content**;
- flex-basis: **min-content**;
- flex-basis: **fit-content**;
- flex-basis: **content**;

Podemos combinar **flex-grow**, **flex-shrink** y **flex-basis** en una única propiedad: **flex**.

Ejemplos:

- **flex: <grow> <shrink> <basis>;**
- **flex: 1 2 300px;**
- **flex: 2 0 5em;**
- **flex: 1 1 95%;**

LINKS

LINKS

- **Flex-box**

- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://flexbox.malven.co/>
- <https://codepen.io/osublake/pen/dMLQJr>
- <https://codepen.io/irinakramer/pen/jcLlp>



[gustavgueez](#)



[gustavgueez](#)

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE