



CSS Flexbox

Modelo unidimensional de layout CSS

https://developer.mozilla.org/es/docs/Web/CSS/CSS_flexible_box_layout



Juan Vladimir
@juanvladimir13

Agenda

1. CSS Flexbox
2. Terminología
3. Container: Propiedades por defecto
4. Container: Propiedades de alineación
5. Items: Propiedades de alineación y orden
6. Container multilinea
7. Espacio entre items

flexbox

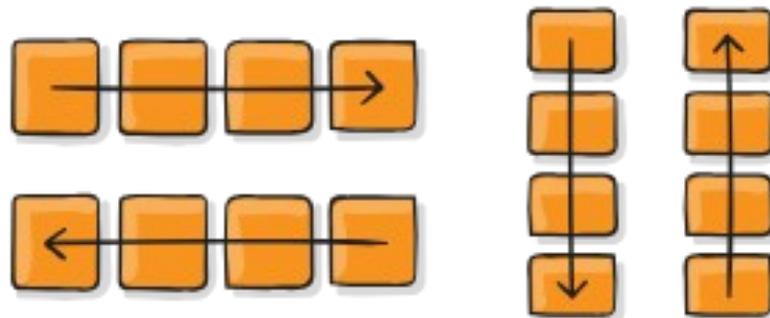
Modelo **unidimensional** de **layout** ya sea como **fila** o como **columna**, un método que ayuda a distribuir el **espacio entre los ítems** de una interfaz y mejorar las **capacidades de alineación**

Eje principal

Propiedad: flex-direction

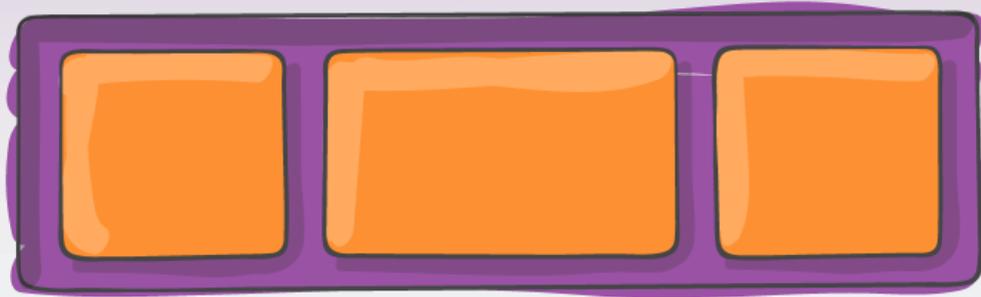
Valor: row | column | row-reverse | column-reverse

```
container-selector {  
  display: flex;  
  flex-direction: row;  
}
```

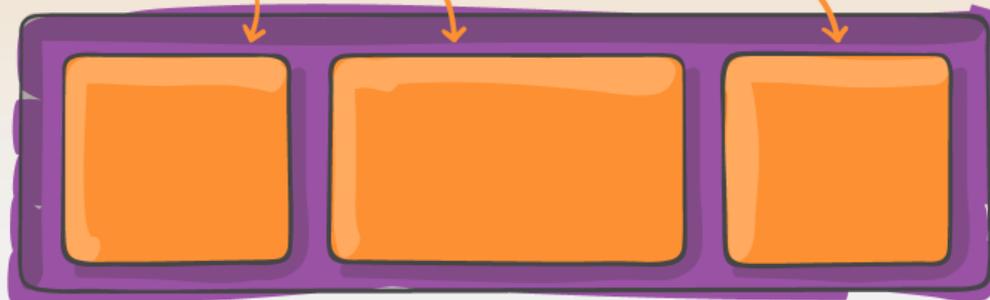


Terminología

□ container



■ items



Container: Propiedades por defecto

Cualquier elemento HTML con la propiedad **display**: flex | inline-flex

Propiedades por defecto:

```
container-selector {  
  display: flex;  
  flex-direction: row;  
  flex-basis: auto;  
  flex-wrap: nowrap;  
}
```

- ✓ Los items empiezan desde el margen inicial sobre el eje principal.
- ✓ Los items no se ajustan en la dirección principal, pero se pueden contraer.
- ✓ Los items se ajustaran para llenar el tamaño del eje cruzado
- ✓ Todos los ítems se alinearán en una sola fila, usando el tamaño del contenedor como su tamaño en el eje principal
- ✓ Si los ítems no caben en el contenedor, estos no pasarán más abajo si no que sobrepasarán el margen
- ✓ Si hay ítems más altos que otros, todos los ítems serán ajustados en el eje cruzado para alcanzar al mayor.

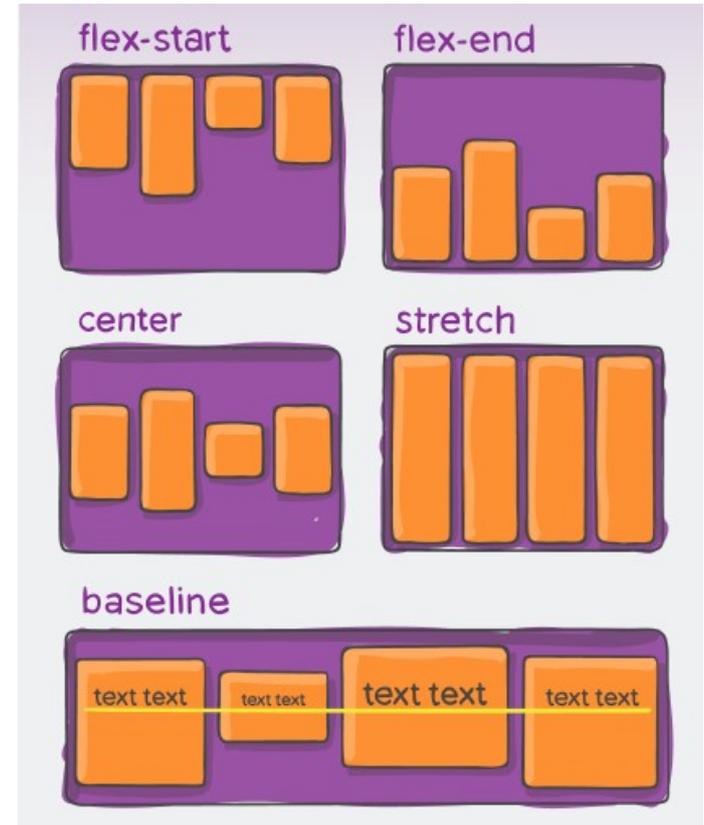
Container: Propiedades de alineación

Propiedad: align-items

Alineación de todos los elementos en el eje principal

Sintaxis:

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



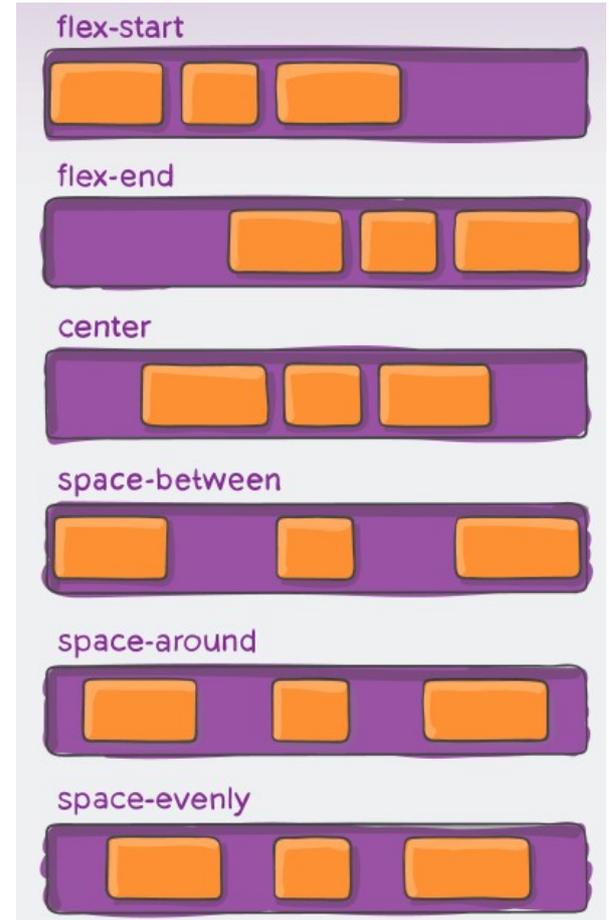
Container: Propiedades de alineación

Propiedad: justify-content

Alineación de todos los elementos en el eje transversal

Sintaxis:

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



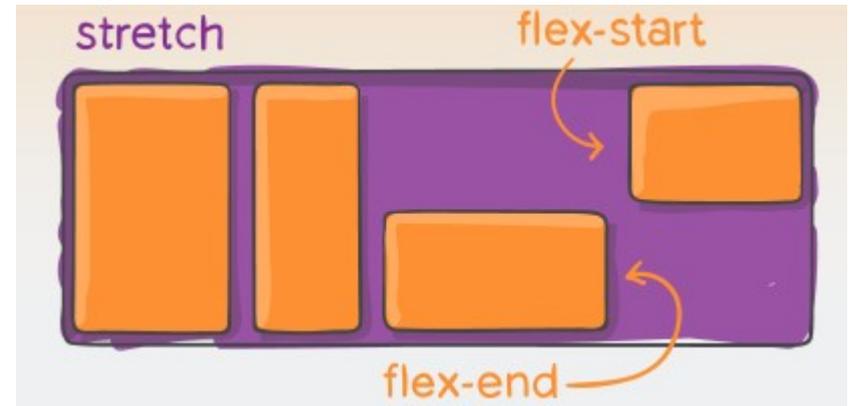
Items: Propiedades de alineación

Propiedad: align-self

Alineación de un elemento flexible individual en el **eje transversal**

Sintaxis:

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



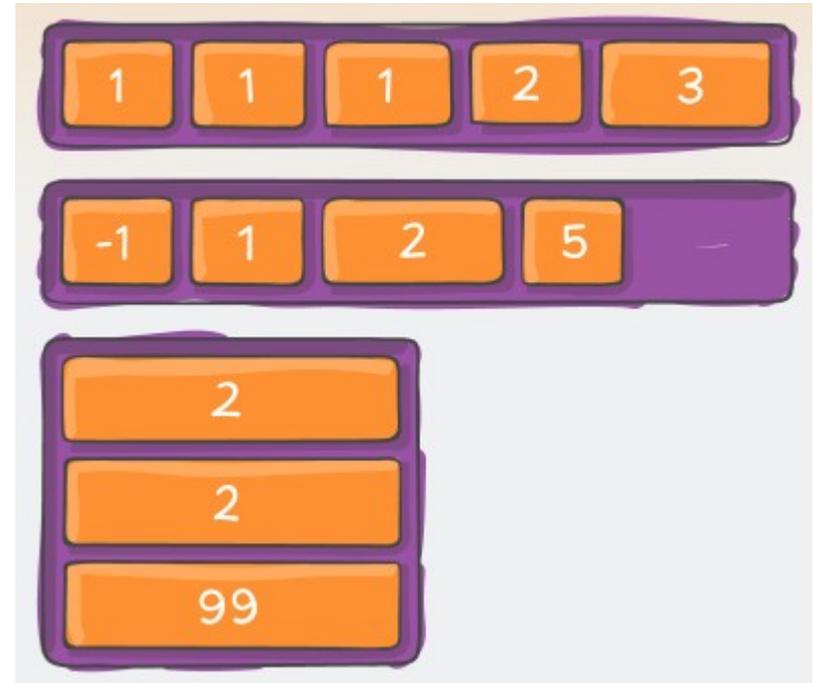
Items: Propiedades de orden

Propiedad: order

Utilizado para disponer el orden de los elementos en su **contenedor**, estarán en orden ascendente según el valor de **order**. Los elementos con el mismo valor se dispondrán en el orden en el cual aparecen en el código fuente.

Sintaxis:

```
.item {  
  order: 5; /* default is 0 */  
}
```



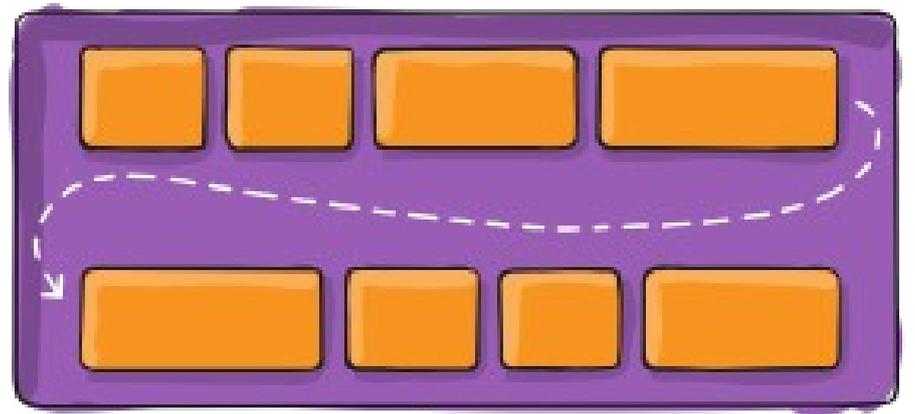
Container multi-line

Cuando los **ítems** son demasiados para desplegarlos en una línea, serán repartidos en la línea siguiente.

Propiedad: flex-wrap

Sintaxis:

```
.container {  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```



gap, row-gap, column-gap

Espacio entre elementos de **fila** o **columna**

```
container-selector {  
  display: flex;  
  flex-wrap: wrap;  
  gap: 15px 10px;  
}
```

```
container-selector {  
  display: flex;  
  flex-wrap: wrap;  
  column-gap: 10px;  
  row-gap: 10px;  
}
```

Propiedades avanzadas

flex-basis: auto | number | fill | max-content | min-content | fit-content | content

Si los ítems no tienen un tamaño, entonces es usado el tamaño de su contenido.

flex-grow: number;

Distribuye el espacio disponible en el **contenedor** de acuerdo al valor que tengan los **items**.

flex-shrink: number;

Controla como se **contrae un ítem** si no contamos con **suficiente espacio en el contenedor**. El ítem puede contraerse a partir de **flex-basis**, un ítem con un valor más alto se contraerá más rápido que sus hermanos que poseen valores menores.

<https://developer.mozilla.org/es/docs/Web/CSS/flex-grow>

<https://developer.mozilla.org/es/docs/Web/CSS/flex-shrink>