



Diseño de Interfaces Web

Flex

Contenidos



Accesibilidad

ÍNDICE

Introducción.....	3
Flexbox.....	3
<i>Elementos básicos.....</i>	<i>3</i>
<i>Resumen propiedades.....</i>	<i>3</i>
<i>Activar modo Flex.....</i>	<i>4</i>
<i>Dirección de los ejes.....</i>	<i>4</i>
<i>Desbordamiento.....</i>	<i>5</i>
<i>Huecos.....</i>	<i>6</i>
Alineación de Elementos.....	7
<i>Propiedades de alineación.....</i>	<i>7</i>
<i>justify-contens vs align-items.....</i>	<i>8</i>
<i>Propiedades de alineación (varias líneas).....</i>	<i>9</i>
<i>Propiedades de alineación (sobre un items).....</i>	<i>10</i>
Propiedades de hijos.....	11
<i>Flexibilidad de elementos hijos.....</i>	<i>11</i>
<i>Orden de elementos hijos.....</i>	<i>12</i>
bibliografía.....	13

1. INTRODUCCIÓN

En la actualidad se utilizan las técnicas de **Flex** y **Grid** para el posicionamiento web, complementando a: *position, fixed, absolute, relative, static*

- **Flex** → Maneja el layout *en una sola dimensión a la vez*, ya sea como fila o como columna.
- **Grid** → Maneja el layout en *2 dimensiones*, controlando las columnas y filas a la vez.

2. FLEXBOX

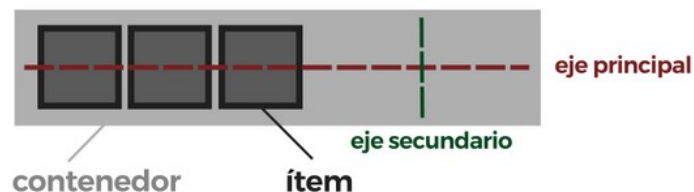
2.1 Elementos básicos

Los elementos básicos en Flex son el **contenedor** y los **ítems**

CONTENEDOR: Es el elemento padre que tendrá en su interior cada uno de los ítems flexibles. Observa que al contrario que muchas otras estructuras CSS, por norma general, en Flex establecemos las propiedades al elemento padre.

- **Eje principal** → Los contenedores flexibles tendrán una orientación principal específica.
 - Por defecto, es en horizontal (en fila).
- **Eje secundario** → De la misma forma, los contenedores flexibles tendrán una orientación secundaria, perpendicular a la principal.
 - Si la principal es en horizontal, la secundaria será en vertical, y viceversa.

ITEM : Cada uno de los hijos flexibles que tendrá el contenedor en su interior.



2.2 Resumen propiedades

Atributo	Descripción
flex-direction	Orientación del eje principal <i>row row-reverse column column-reverse</i>
flex-wrap	Comportamiento si hay desbordamiento de los ítem <i>wrap no-wrap</i>
justify-content	Alinear los ítems por el eje principal <i>flex-start flex-end center space-between space-around space-evenly</i>
align-items	Alinear los ítems por el eje secundario <i>flex-start flex-end center stretch baseline</i>
row-gap	Espacio entre filas <i>normal size</i>
column-gap	Espacio entre columnas <i>normal size</i>
Atributo	Descripción
flex-grow	Factor de crecimiento <i>0 número</i>
flex-shrink	Factor de decrecimiento <i>1 número</i>
flex-basis	Tamaño básico inicial <i>tamaño content</i>
flex-order	Orden de aparición <i>0 número</i>

2.3 Activar modo Flex

Para activar el modo flexbox hay que utilizar la propiedad **display** en el contenedor y especificar el valor **flex** o **inline-flex**:

***Dependiendo de como queramos que se comporte el contenedor*

flex	El contenedor se comporta como elemento en bloque → Ocupa solo el espacio de su contenido
flex-inline	El contenedor se comporta como elemento en línea → Ocupa todo el ancho del padre

```
<div class="contenedor">                                <!-- Conenedor flex-->
  <div class="item item-1">1</div>                        <!-- items flex -->
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
</div>
```

```
.contenedor {
  display: flex; /* display: inline-flex */
}
```



2.4 Dirección de los ejes

Existen dos propiedades principales para manipular la dirección y comportamiento de los ítems a lo largo del eje principal del contenedor. Son las siguientes:

- flex-direction** → Cambia la orientación del eje principal.
- flex-wrap** → Evita o permite el desbordamiento (multilinea).
- flex-flow** → Atajo para las dos propiedades anteriores.

flex-direction

Modificar la dirección del **eje principal** del contenedor para que sea horizontal o vertical.

- Incluir el sufijo **-reverse** para indicar que coloque los ítems en orden inverso.

row	Establece la dirección del eje principal en horizontal.
row-reverse	Establece la dirección del eje principal en horizontal (invertido).
column	Establece la dirección del eje principal en vertical.
column-reverse	Establece la dirección del eje principal en vertical (invertido).



2.5 Desbordamiento

flex-wrap

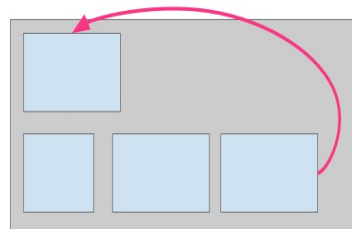
Especificar el comportamiento del contenedor cuando los ítems no caben en una línea y se desbordan. Puede tener los valores:

nowrap	Establece los ítems en una sola línea (<i>no permite que se desborde el contenedor</i>).
wrap	Establece los ítems en modo multilinea (<i>permite que se desborde el contenedor</i>).
wrap-reverse	Establece los ítems en modo multilinea, pero en dirección inversa.

```
.contenedor {  
  display: flex;  
  flex-wrap: nowrap;  
}  
.item {  
  width: 50%;  
}
```



wrap

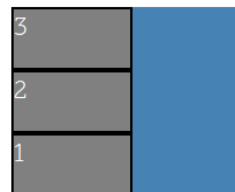
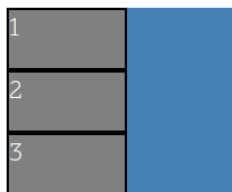
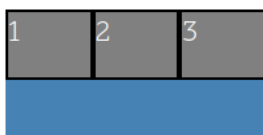


wrap-reverse

Aclaración

nowrap → Los 3 ítems se mostrarían en una misma línea del contenedor. Cada ítem debería tener un 50% de ancho (o sea, 100px de los 200px del contenedor). Un tamaño de **100px** por ítem, sumaría un total de **300px**, que no cabrían en el contenedor de **200px**, por lo que **flexbox** reajusta los ítems flexibles para que quepan todos en la misma línea, manteniendo las mismas proporciones.

wrap → Permitimos que el contenedor se pueda desbordar, pasando a ser un contenedor **multilinea**, que mostraría el **ítem 1** y **2** en la primera línea (con un tamaño de 100px cada uno) y el **ítem 3** en la línea siguiente, dejando un espacio libre para un posible **ítem 4**.



flex-flow [Atajo]

Propiedad de atajo para resumir los valores de las propiedades: flex-direction y flex-wrap,

```
flex-flow: <flex-direction> <flex-wrap>
```

```
.contenedor {  
  display: flex  
  flex-flow: row wrap  
}
```

2.6 Huecos

Permiten establecer el tamaño de un «hueco» entre ítems desde el elemento padre contenedor, y sin necesidad de estar utilizando padding o margin en los elementos hijos.

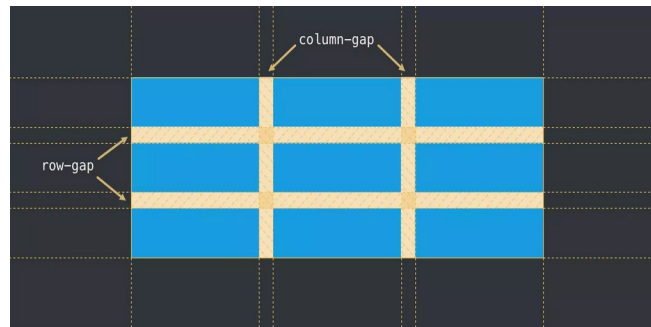
- **row-gap**
- **column-gap**.

row-gap	<i>normal size</i>	<i>Espacio entre filas (sólo si flex-direction: column)</i>
column-gap	<i>normal size</i>	<i>Espacio entre columnas (sólo si flex-direction: row)</i>



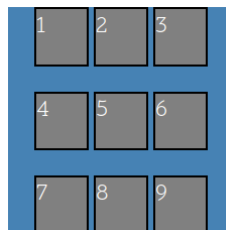
ACLARACIÓN → Flex es un sistema para diseños de una sola dimensión, sólo una de las dos propiedades tendrá efecto.

Es posible usar ambas si tenemos la propiedad flex-wrap definida a wrap



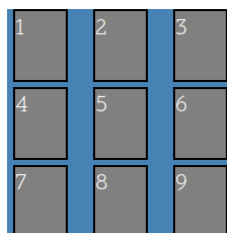
row-gap

```
.contenedor {  
  justify-content: center ;  
  flex-wrap: wrap;  
  row-gap: normal  
}
```



column-gap

```
.contenedor {  
  justify-content: center ;  
  flex-wrap: wrap;  
  column-gap: normal  
}
```



gap [atajo]

propiedad de atajo sobre el contenedor padre para resumir los valores de las propiedades: “row-gap” y “column-gap”

```
gap: <row> <column>
```

```
.contenedor {  
  display: flex  
  gap: 4px 8px  
  gap: 4px  
}
```

// El mismo en ambos

3. ALINEACIÓN DE ELEMENTOS

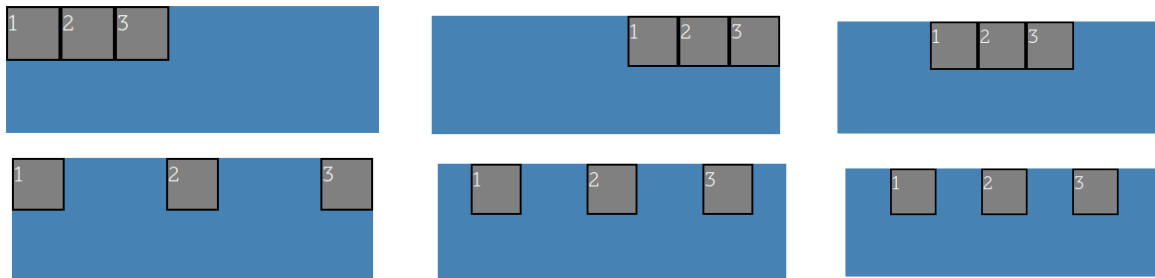
3.1 Propiedades de alineación

justify-content	Alinear los ítems por el eje principal	
align-items	Alinear los ítems por el eje secundario	
align-content	Alinear elementos cuando son varias líneas	MULTILINEA(WRAP)
align-self	Alinear elemento de forma individual	ELEMENTO

justify-content

Colocar los ítems de un contenedor mediante una disposición concreta a lo largo del eje principal:

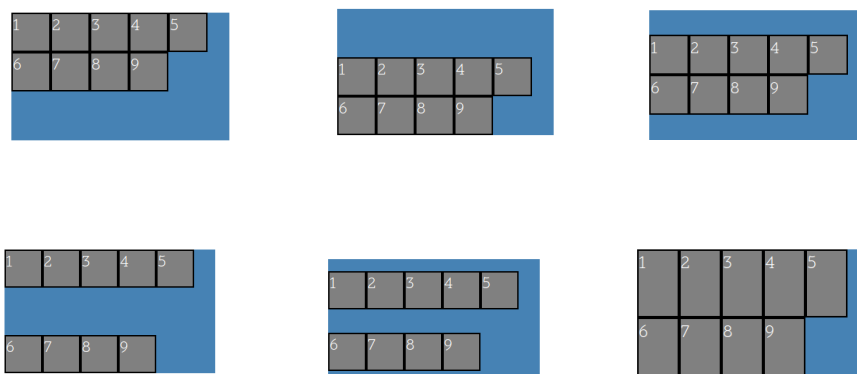
start	Agrupar los ítems al principio
end	Agrupar los ítems al final
center	Agrupar los ítems al centro
space-between	Distribuye los ítems dejando el máximo espacio para separarlos
space-around	Distribuye los ítems dejando el mismo espacio alrededor de ellos (izq/dcha).
space-evenly	Distribuye los ítems dejando el mismo espacio (solapado) a izquierda y derecha.



align-items

colocar los ítems de un contenedor mediante una disposición concreta a lo largo del eje secundario:

start	Agrupar los ítems al principio
end	Agrupar los ítems al final
center	Agrupar los ítems al centro
stretch	Distribuye los ítems dejando el máximo espacio para separarlos
baseline	Distribuye los ítems dejando el mismo espacio alrededor de ellos (izq/dcha).



3.2 justify-content vs align-items

	flex-start	flex-end	center	stretch	baseline
flex-start					
flex-end					
center					
space-between					
space-around					
space-evenly					

place-content [Atajo]

Esta propiedad es un atajo con la que se establecen los valores de las propiedades anteriores : justify-content y align-items

place-content: <justify-content> <align-items>

```
.contenedor {
  display: flex
  place-content: start end
}
```


3.3 Propiedades de alineación (varias líneas)

align-content	Alinear elementos cuando son varias líneas
----------------------	--

align-content

Colocar los ítems cuando están formando varias líneas:

start	Agrupar los ítems al principio
end	Agrupar los ítems al final
center	Agrupar los ítems al centro
stretch	Distribuye los ítems dejando el máximo espacio para separarlos
space-around	Distribuye los ítems dejando el mismo espacio alrededor de ellos (izq/dcha).
space-evenly	Distribuye los ítems dejando el mismo espacio (solapado) a izquierda y derecha.

```
.contenedor {  
  align-items : stretch  
  align-content : flex-end  
}
```

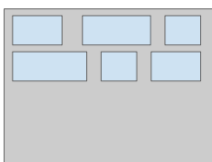
Aclaración



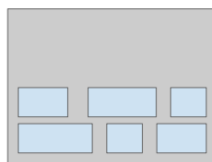
Ambas reglas (align-items y align-content), no se solapan.

- Si los elementos forman **una línea** se comportan según la propiedad **align-items**.
- Si los elementos forman **varias líneas** se comportan según la propiedad **align-content**.

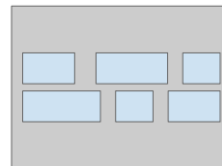
flex-start



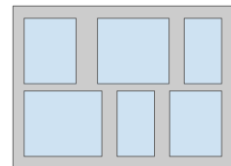
flex-end



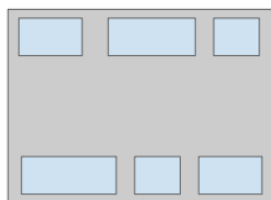
center



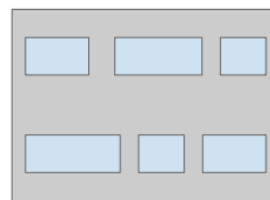
stretch (no height)



space-between



space-around



3.4 Propiedades de alineación (sobre un items)

En ocasiones puedo necesitar que un elemento flexible tenga una alineación vertical diferente al resto. En este caso, en el elemento para el que quiero una alineación diferente, debo añadir la propiedad **align-self** que puede tomar los mismo valores (y con el mismo significado) que la propiedad **align-items**.

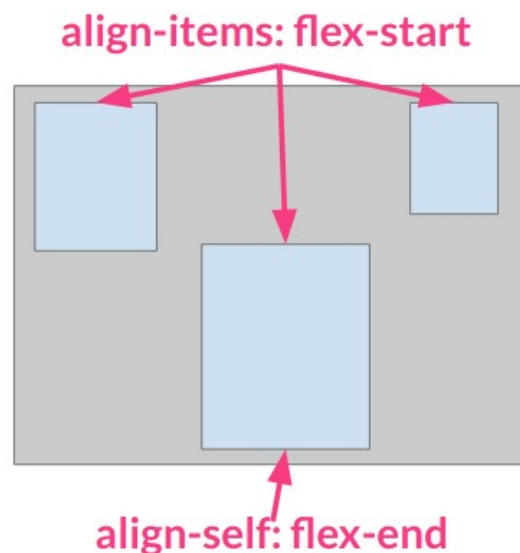
align-self	<i>Alinear elemento de forma individual</i>
-------------------	---

align-self

Colocar los ítems cuando están formando varias líneas:

start	<i>Alinear ítem al principio</i>
end	<i>Alinear ítem al final</i>
center	<i>Centrar ítem</i>
stretch	<i>No debe tener altura establecida</i>
baseline	<i>Distribuye el ítem dejando el mismo espacio alrededor de él (izq/dcha).</i>

```
.item {  
  align-self : flex-start  
}
```



4. PROPIEDADES DE HIJOS

4.1 Flexibilidad de elementos hijos

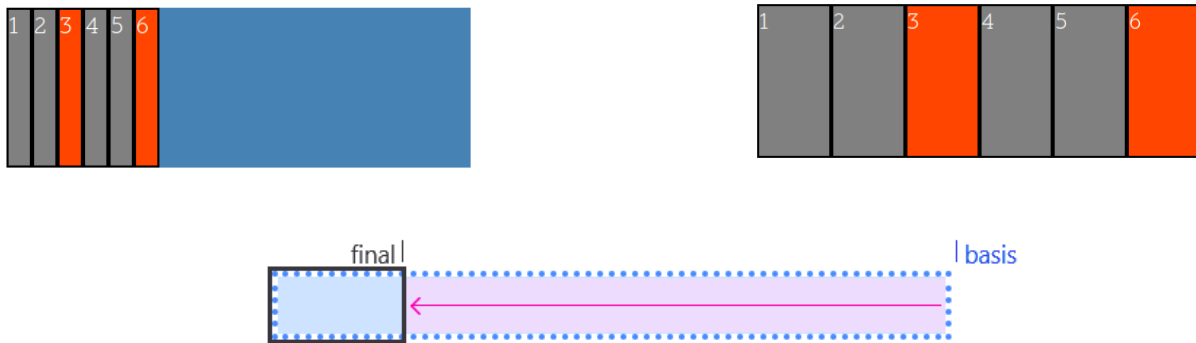
Todas las propiedades que hemos visto hasta ahora se aplican sobre el elemento **contenedor**. Las siguientes propiedades, sin embargo, se aplican sobre los ítems hijos. Echemos un vistazo:

flex-basis	<i>tamaño content</i>	<i>Tamaño base de los ítems antes de aplicar variación.</i>
flex-grow	<i>0 número</i>	<i>Factor de crecimiento del ítem respecto al resto.</i>
flex-shrink	<i>1 número</i>	<i>Factor de decrecimiento del ítem respecto al resto.</i>
order	<i>0 número</i>	<i>Número (peso) que indica el orden de aparición de los ítems.</i>

flex-basis

Define el tamaño base por defecto que tendrán los ítems antes de aplicarle una cierta distribución de espacio.

- Se suele aplicar un tamaño específico (unidades, porcentajes, etc...).
- Se puede ajustar automáticamente al tamaño del contenido del elemento, mediante el valor "content".



flex-grow

Esta propiedad actúa en situaciones donde:

- Hay un flex-basis definido.
- Los ítems cubren el tamaño total del contenedor flex padre.

En esas situaciones, la propiedad flex-grow, indica el factor de crecimiento de los ítems.

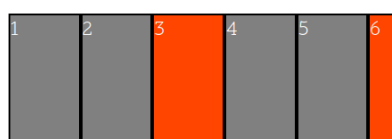


flex-shrink

Por otro lado, tenemos la propiedad flex-shrink que es la opuesta a la propiedad flex-grow. Mientras que la anterior indica un factor de crecimiento, flex-shrink, hace justo lo contrario: aplica **factor de decrecimiento**.

Esta propiedad actúa en situaciones donde:

- Hay un flex-basis definido.
- Los ítems **no** cubren el tamaño total del contenedor flex padre.



flex [atajo]

flex:

propiedad de atajo sobre los hijos para resumir los valores de las propiedades: “*flex-basic*” y “*flex-grow*”, “*flex-shrink*”

```
flex: <flex-grow> <flex-shrink> <flex-basic>
```

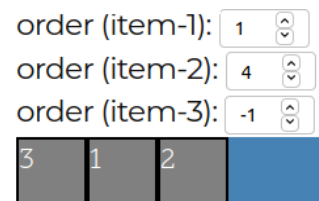
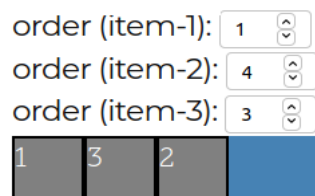
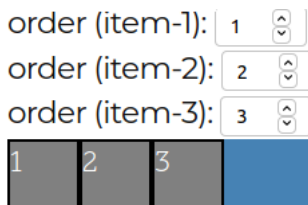
```
.item {  
  display: flex  
  flex: 1 3 35%  
}
```

4.2 Orden de elementos hijos

Modificar y establece el orden de los ítems según una secuencia numérica.

Todos los ítems flex tienen un orden: 0 implícito, aunque no se especifique. Si indicamos un orden con un valor numérico, irá recolocando los ítems según su número, colocando antes los ítems con número más pequeño (*incluso valores negativos*) y después los ítems con números más altos.

```
.contenedor {  
  display: flex  
}  
.item1 { order: 1 }  
.item2 { order: 2 }  
.item3 { order: 3 }
```



5. BIBLIOGRAFÍA

BIBLIOGRAFÍA



- **Web personal Manz**
 - <https://lenguajecss.com/css/maquetacion-y-colocacion/flex/>
- **Mozilla Developer**
 - <https://developer.mozilla.org/es/docs/Web/CSS/flex>
- **Wikipedia**
 - https://es.wikipedia.org/wiki/CSS_Flexible_Box_Layout
- **OpenWebinar**
 - <https://openwebinars.net/academia/aprende/flexbox-css-grid/>
- **Generador automático**
 - <https://layout.bradwoods.io/customize>
- **Web personal Marina Ferreira**
 - <https://marina-ferreira.github.io/tutorials/css/flexbox/>

