



## *Diseño de Interfaces Web*

### UD – 4

#### *Contenido Interactivo*

---

## Contenidos

---



**Accesibilidad**

# ÍNDICE

<b>Introducción.....</b>	<b>3</b>
<b>Flexbox.....</b>	<b>3</b>
<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>Propiedades de alineación.....</i>	<i>6</i>
<i>justify-content vs align-items.....</i>	<i>7</i>
<i>Propiedades de alineación (varias líneas).....</i>	<i>8</i>
<i>Propiedades de alineación (sobre un items).....</i>	<i>9</i>
<i>Huecos.....</i>	<i>10</i>
<b>Propiedades de hijos.....</b>	<b>11</b>
<i>Orden de Items.....</i>	<i>13</i>

# 1. INTRODUCCIÓN

En la actualidad se utilizan las técnicas de **Flex** y **Grid** para el posicionamiento web, sustituyendo 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

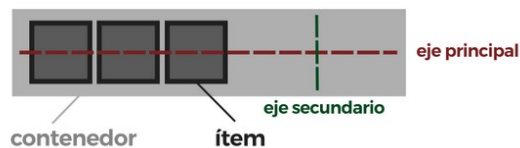
### 2.1 Elementos básicos

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

<b>flex</b>	El contenedor se comporta como <b>elemento en bloque</b>
<b>flex-inline</b>	El contenedor se comporta como <b>elemento en línea</b>

```
<div class="container">
  <div class="item item-1">1</div>
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
</div>
```

*<!-- Conenedor flex-->*  
*<!-- items flex -->*

```
.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.*

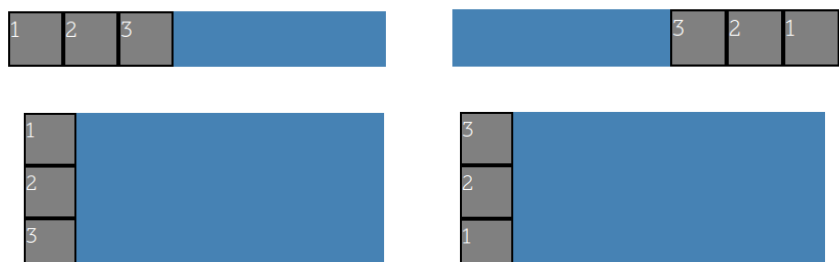
### flex-direction

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.*

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

```
.contenedor {
  display: flex;
  flex-direction: row
}
```



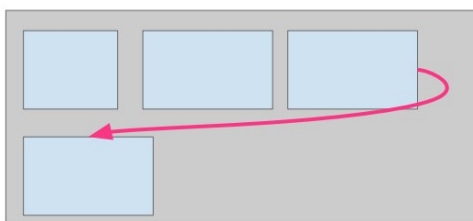
## 2.5 Desbordamiento

### flex-wrap

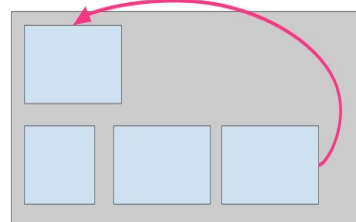
Especificar el comportamiento del contenedor cuando los items no caben en una línea y se desbordan. Puede tener los valores:

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

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



wrap

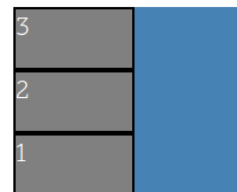
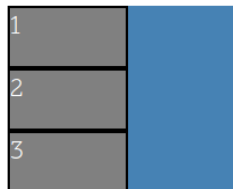


wrap-reverse

### 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  
}
```

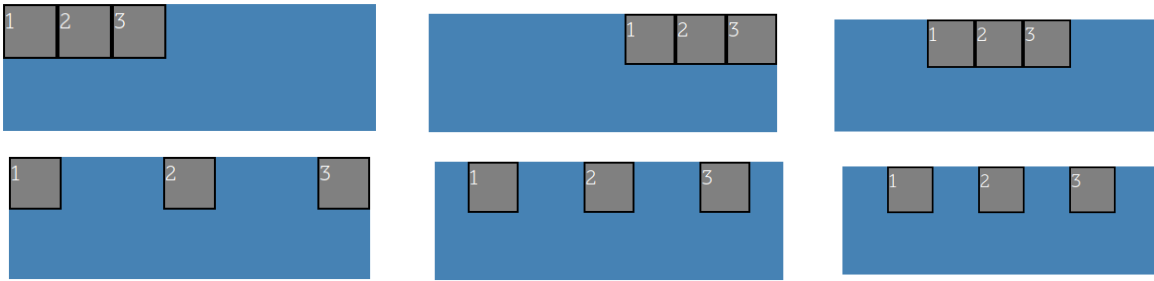
## 2.6 Propiedades de alineación

<b>justify-content</b>	<i>Alinear los ítems por el eje principal</i>
<b>align-items</b>	<i>Alinear los ítems por el eje secundario</i>

### justify-content

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

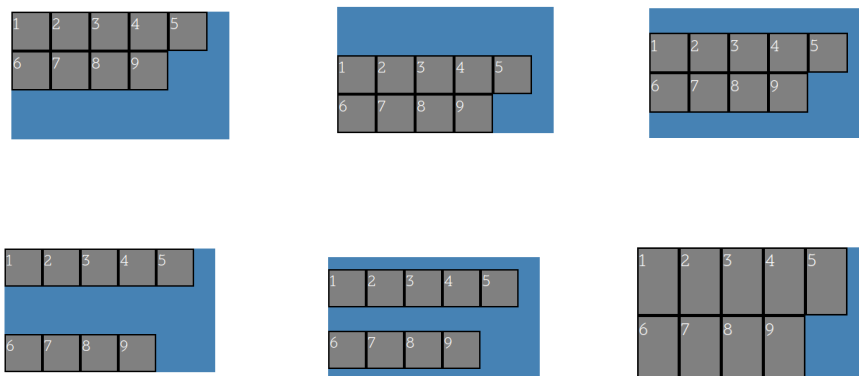
<b>flex-start</b>	<i>Agrupar los ítems al principio</i>
<b>flex-end</b>	<i>Agrupar los ítems al final</i>
<b>center</b>	<i>Agrupar los ítems al centro</i>
<b>space-between</b>	<i>Distribuye los ítems dejando el máximo espacio para separarlos</i>
<b>space-around</b>	<i>Distribuye los ítems dejando el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



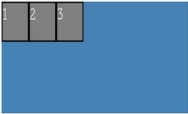

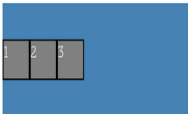




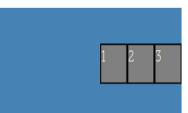


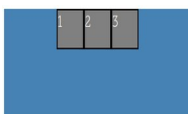
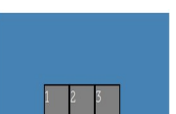
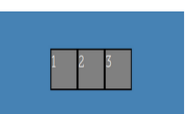
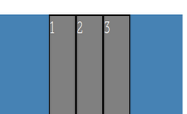
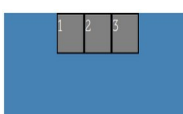
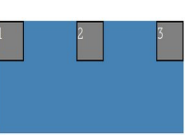
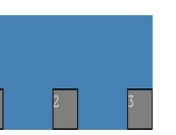
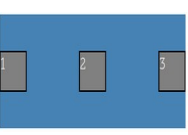
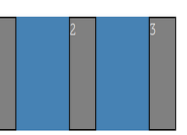
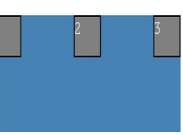
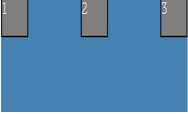
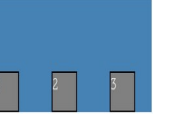
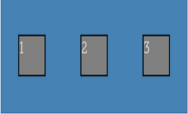
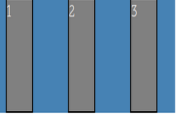
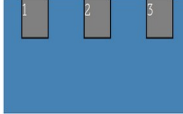
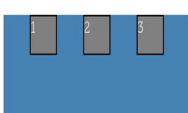

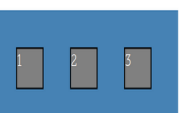
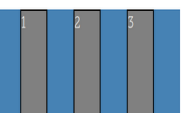

### align-items

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

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



## 2.7 justify-contens vs align-items

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

## 2.8 Propiedades de alineación (varias líneas)

<b>align-content</b>	Alinear elementos cuando son varias líneas
----------------------	--

### align-content

Colocar los ítems cuando están formando varias líneas:

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

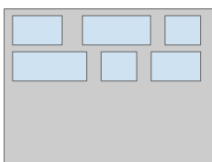
```
.contenedor {  
  align-items : stretch  
  align-content : flex-end  
}
```

### Aclaración

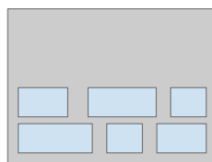
Ambas reglas (align-items y align-content), no se solapan.

- 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**.

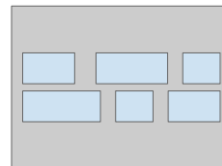
flex-start



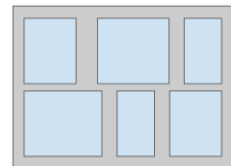
flex-end



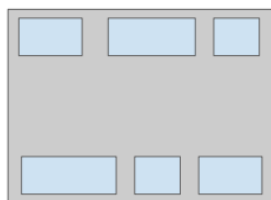
center



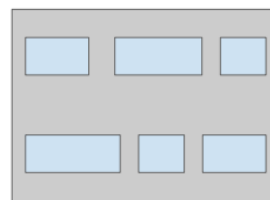
stretch (no height)



space-between



space-around





## 2.9 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 CSS **align-self** que puede tomar los mismo valores (y con el mismo significado) que la propiedad **align-items**.

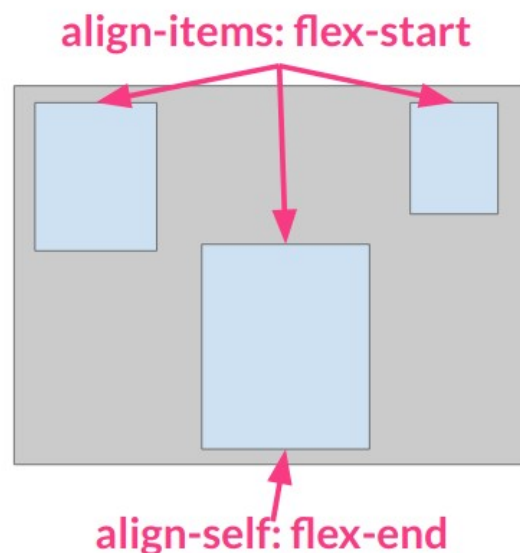
<b>align-self</b>	<i>Alinear elemento de forma individual</i>
-------------------	---

### align-self

Colocar los ítems cuando están formando varias líneas:

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

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



## 2.10 Huecos

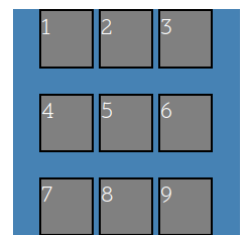
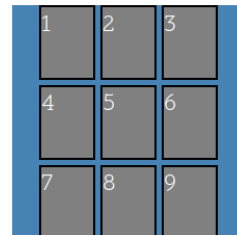
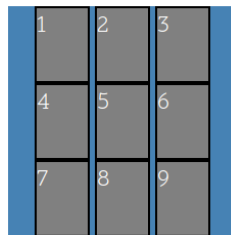
Permiten establecer el tamaño de un «hueco» entre ítems desde el elemento padre contenedor, y sin necesidad de estar utilizando padding o margin en los elementos hijos.

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

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

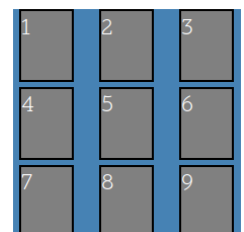
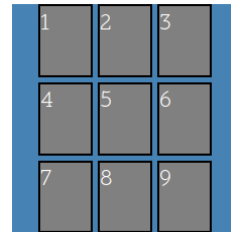
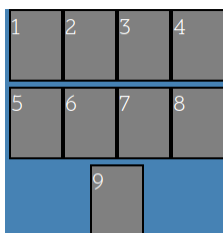
### row-gap

```
.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  
}
```

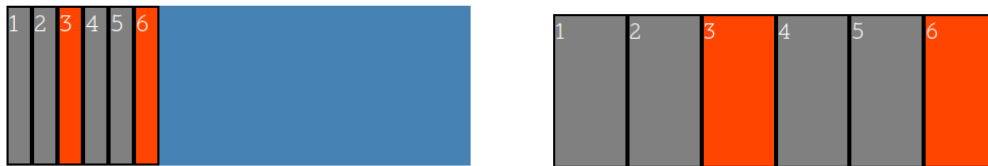
### 3. PROPIEDADES DE HIJOS

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:

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

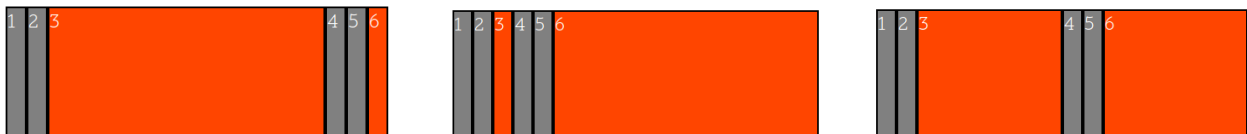
#### flex-basic

Tamaño de un elemento antes de que el espacio restante (negativo o positivo) se distribuye. (auto por defecto)



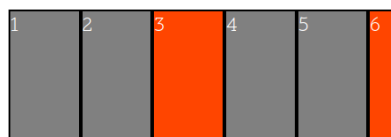
#### flex-grow

El factor de crecimiento de un elemento flexible cuando se distribuye el espacio restante. Por defecto 1 pero debo indicarlo en todos.



#### flex-shrink

El factor de contracción de un elemento flexible cuando sobrepasan el tamaño del contenedor. Por defecto 1 pero debo indicarlo en todos.



#### 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%  
}
```

### 3.1 Orden de Items

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 }
```

order (item-1):

order (item-2):

order (item-3):

1	2	3	
---	---	---	--

order (item-1):

order (item-2):

order (item-3):

1	3	2	
---	---	---	--

order (item-1):

order (item-2):

order (item-3):

3	1	2	
---	---	---	--