

### FLEXBOX (Flexible Box Layout)

Es un método de diseño unidimensional en CSS que resulta flexible para organizar y distribuir elementos. A diferencia del Grid Layout, que permite la organización bidimensional, Flexbox se centra en la distribución unidimensional. Por esta razón, es habitual utilizarlo para la creación de barras de navegación y galerías de imágenes, entre otros componentes, puesto que facilita la alineación vertical y horizontal de elementos. Algunas ventajas:

- **Simplifica la maquetación:** elimina la necesidad de flotar elementos.
- **Control del espacio:** incorpora propiedades que permiten un control preciso del espacio entre los elementos, tanto en el nivel horizontal como en el vertical, lo que facilita la creación de diseños limpios y agradables.
- **Responsivo:** está especialmente destinado a la realización de diseños que se adapten fácilmente a diferentes tamaños de pantalla y dispositivos, mejorando la experiencia del usuario.
- **Ordenación de elemntos:** permite cambiar el orden visual de los elementos sin variar el orden enel HTML, lo que es útil para presental la información de manera efectiva.

#### CONTENEDOR FLEX Y ELEMENTOS FLEX

Un contenedor Flex es un elemento HTML que tiene sus hijos directos configurados para usar Flexbox. Al habilitar un contenedor como Flex, este se convierte en un contexto de diseño flexible sobre el que distribuir el resto de los elementos que dependen de él.

El contenedor debe incluir **display: flex;** a partir de ahí sus elementos hijos pueden ser colocados en filas o columnas.

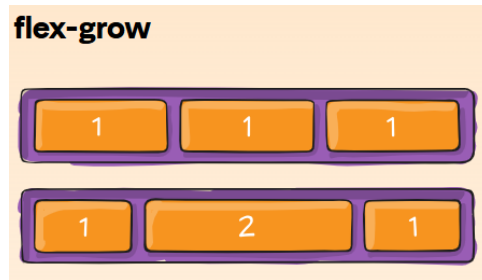
```
.contenedor {  
  display: flex;  
}
```



Los elementos que son hijos directos de un contenedor Flex serán los elementos flexibles que participan en el diseño. Las propiedades de los elementos Flexbox serán definidas mediante la propiedad abreviada **flex** o las propiedades individuales **flex-grow**, **flex-shrink** y **flex-basis**

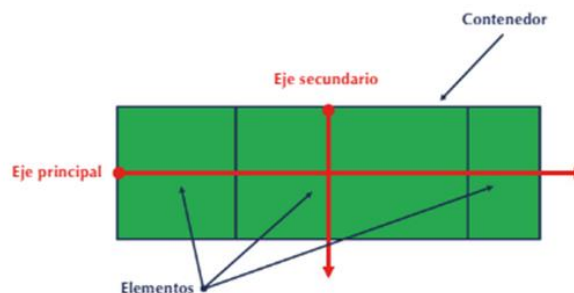
```
.item {  
  flex: flex-grow flex-shrink flex-basis  
}
```

- **flex-grow:** define el espacio adicional que debe tomar el elemento en relación con los otros elementos dentro del mismo contenedor. Un valor de 1 significa que tomará todo el espacio adicional disponible.
- **flex-shrink:** define la capacidad del elemento para reducir su tamaño, en caso de ser necesario para evitar el desbordamiento del contenedor.
- **flex-basis:** define el tamaño inicial del elemento antes de que se distribuya el espacio adicional o se apliquen los factores de reducción.



### EJES: PRINCIPAL Y TRANSVERSAL

Los ejes son los elementos fundamentales en Flexbox. Se refieren a las direcciones sobre las cuales se distribuirán y alinearán los elementos flexibles. Hay dos ejes esenciales: el eje principal (main axis) y el eje secundario (cross axis). Estos dos ejes son importantes para saber cómo se comportan y distribuyen los elementos flexibles dentro de un contenedor flex.



El eje principal es la dirección en la que se colocan los elementos flexibles dentro del contenedor. Para indicar cuál de los ejes es el principal se utiliza la propiedad `flex-direction`, seguida de `row` para el caso de la dirección horizontal (como una fila) o `column` para el caso vertical (como una columna). Si no se define ninguno, por defecto será horizontal.

```
.contenedor {  
  flex-direction: row | column;  
}
```

El eje transversal será, en cada caso. El eje perpendicular al eje principal. Es decir, si el eje principal es horizontal, el transversal es vertical y viceversa.

En función de los ejes definidos, se aplican las propiedades de dirección o alineación para los elementos que dependen del contenedor flexbox.

### PROPIEDADES FUNDAMENTALES SOBRE LA DIRECCIÓN

- **direction:** define la dirección de escritura sobre el eje principal, es decir, la dirección en la que se escribe el texto.

```
.contenedor {  
  direction: ltr; /* de izquierda a derecha (por defecto) */  
  direction: rtl; /* de derecha a izquierda */  
}
```

## UT4.-Flexbox

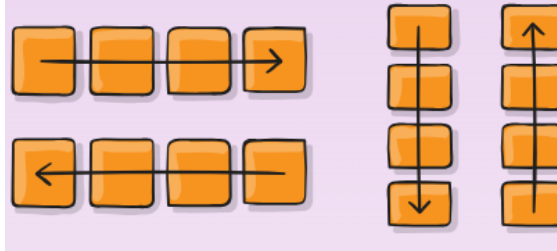
- **flex-direction:** define la dirección en la que se colocan los elementos dentro del contenedor flexible. Puede tomar los siguientes valores: **row** (fila en sentido izquierda-derecha), **row-reverse** (fila en sentido inverso: derecha-izquierda), **column** (columna de arriba abajo) y **column-reverse** (columna sentido inverso, de abajo arriba).

**.contenedor {**

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

**}**

### flex-direction



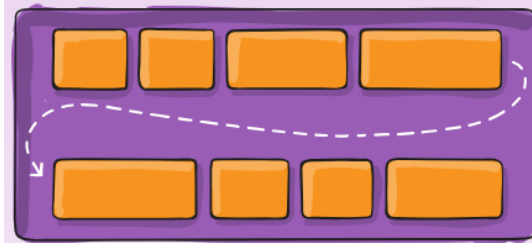
- **flex-wrap:** permite definir si los elementos se van a distribuir en varias líneas, en caso de que el espacio sea insuficiente. Puede tomar los siguientes valores: **wrap** (se pueden distribuir en varias líneas), **nowrap** (no se podrán distribuir en varias líneas) y **wrap-reverse** (se distribuirán en varias líneas, pero en sentido inverso).

**.contenedor {**

**flex-wrap: wrap | nowrap | wrap-reverse;**

**}**

### flex-wrap



- **flex-flow:** propiedad que combina flex-direction y flex-wrap, por lo que toma los valores descritos en cada una de las propiedades.

**.contenedor {**

**flex-flow: row wrap;**

**}**

## PROPIEDADES FUNDAMENTALES SOBRE LA ALINEACIÓN

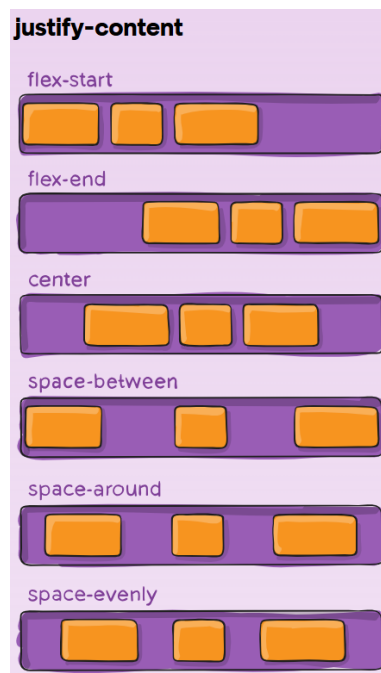
La alineación permite definir cómo se posicionan y ajustan los elementos en el interior de un contenedor flexible, tanto en el eje principal como en el secundario. Se utilizan varias propiedades para controlar la alineación de los elementos:

- **justify-content:** define la alineación de los elementos a lo largo del eje principal del contenedor.

```
.contenedor {
  justify-content: opciones;
}
```

Opciones de justify-content:

<b>flex-start</b>	Los elementos se alinean hacia el inicio del contenedor. El primer elemento estará en la esquina inicial, y los demás seguirán la dirección del eje principal
<b>flex-end</b>	Los elementos se alinean hacia el final del contenedor. El último elemento estará en la esquina final, y los demás se colocarán antes
<b>center</b>	Los elementos se centran dentro del contenedor a lo largo del eje principal
<b>space-between</b>	Los elementos se distribuyen uniformemente a lo largo del eje principal. El primer elemento se coloca al inicio; el último, al final, y el espacio restante se distribuye igualmente entre los elementos
<b>space-around</b>	Similar a <i>space-between</i> , pero con espacio adicional antes del primer elemento y después del último elemento



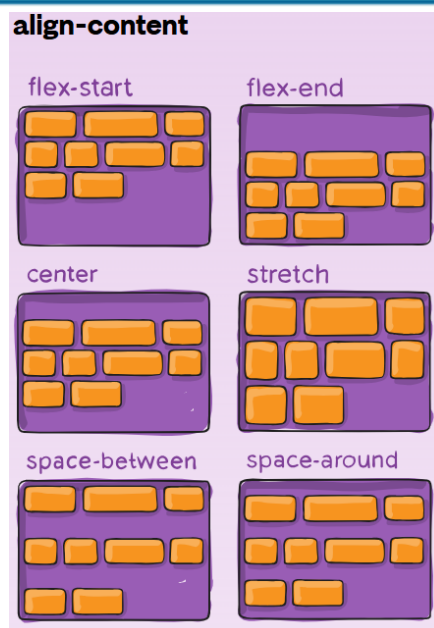
- **align-content:** define cómo se distribuyen el espacio adicional en el eje transversal si hay varias líneas de elementos. Esta propiedad se aplica cuando hay un wrap y afecta al espacio entre filas.

```
.contenedor {
  align-content: opciones;
}
```

## UT4.-Flexbox

Opciones de align-content:

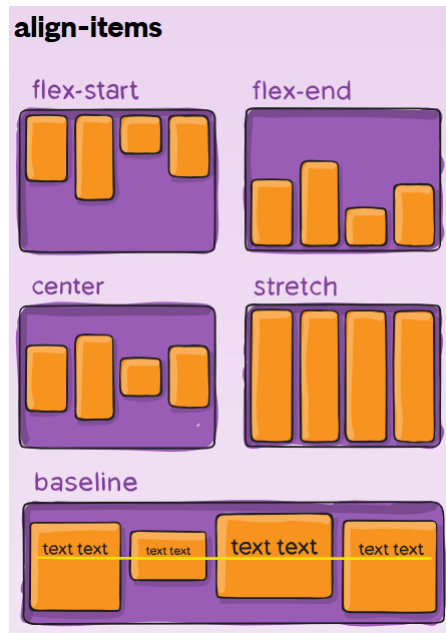
<b>flex-start</b>	Las filas se alinean hacia el inicio del contenedor
<b>flex-end</b>	Las filas se alinean hacia el final del contenedor
<b>center</b>	Las filas se centran verticalmente dentro del contenedor
<b>space-between</b>	Las filas se distribuyen uniformemente a lo largo del eje transversal
<b>space-around</b>	Las filas se distribuyen uniformemente a lo largo del eje transversal, con espacio adicional alrededor de cada fila
<b>stretch</b>	Las filas se estiran para llenar todo el espacio disponible en el eje transversal



- **align-items:** permite definir la alineación de los elementos a lo largo del eje transversal del contenedor.  
**.contenedor {**  
    **align-items: opciones;**  
**}**

Opciones de align-items:

<b>flex-start</b>	Los elementos se alinean hacia el inicio del contenedor
<b>flex-end</b>	Los elementos se alinean hacia el final del contenedor
<b>center</b>	Los elementos se centran dentro del contenedor a lo largo del eje transversal
<b>baseline</b>	Los elementos se alinean en las líneas base del eje transversal
<b>stretch</b>	Los elementos se estiran para llenar todo el espacio disponible en el eje transversal



## ORDENACIÓN

Los elementos organizados en un contenedor Flexbox podrán cambiar su orden de visualización. Para ello se utiliza la propiedad **order**. De forma predeterminada, todos los elementos tienen un valor de 0, lo que significa que se mostrarán en el orden en el que aparecen en el documento HTML. A través de *order*, *se podrá modificar el orden visual de los elementos sin cambiar la estructura del DOM*.

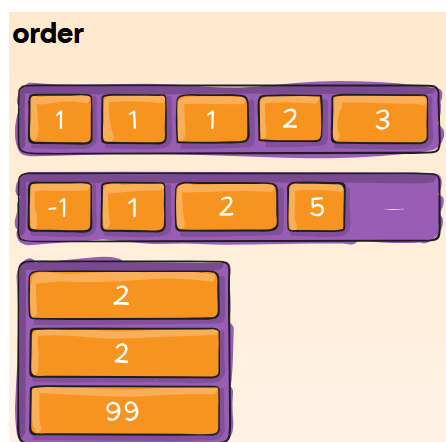
La propiedad **order** recibe como valor la posición en la que se va a colocar el elemento con respecto al resto. Si varios elementos tienen el mismo valor de *order*, se mostrarán en el mismo orden en el que aparecen en la estructura HTML.

**elemento {**

**order: *numero*; /\* numero puede usar números decimales \*/**

**}**

Si se asigna a *order* un número negativo, este elemento se colocará al principio del orden visual, incluso antes que los elementos con *order* valor 0;



## UT4.-Flexbox

Ejemplo:

```
.contenedor-flex {  
  display: flex;  
}  
.elemento-1 {  
  order: 2; /* Se mostraría en la cuarta posición */  
}  
.elemento-2 {  
  order: 1; /* Se mostraría en la tercera posición */  
}  
.elemento-3 {  
  order: 3; /* Se mostraría en la quinta posición */  
}  
.elemento-4 {  
  order: -1; /* Se mostraría en la primera posición */  
}  
.elemento-5 {  
  /* Por defecto, el valor de order es 0 */  
  /* Se mostraría en la segunda posición */  
}
```

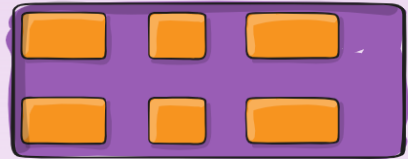
## SEPARACIÓN DE LOS ITEMS

### gap, row-gap, column-gap

gap: 10px



gap: 30px



gap: 10px 30px



```
.container {  
  display: flex;  
  ...  
  gap: 10px;  
  gap: 10px 20px; /* row-gap column gap */  
  row-gap: 10px;  
  column-gap: 20px;  
}
```