

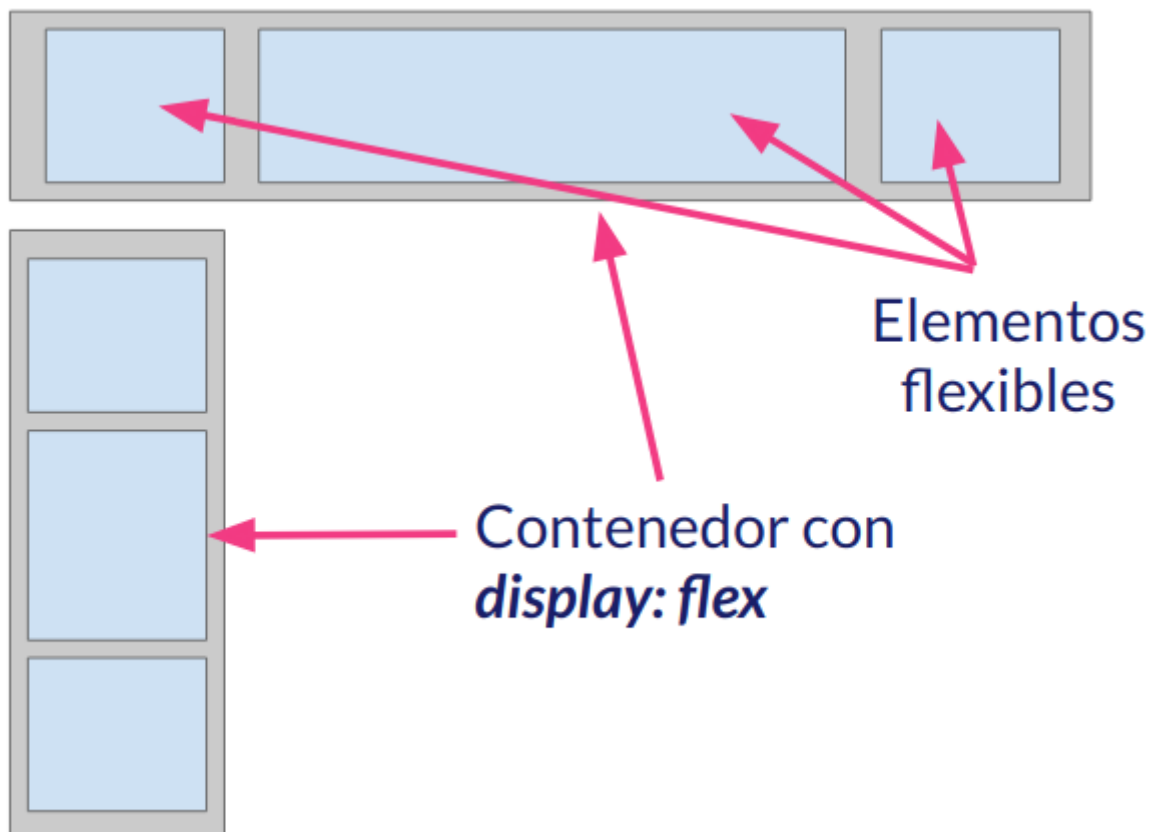
MAQUETACIÓN CON FLEX

1 ELEMENTOS DE MAQUETACIÓN CON FLEX

La principal idea que hay detrás de la maquetación **FLEX** es que vamos a tener un elemento, es decir, una etiqueta que va a poder *controlar* las propiedades de los elementos que contiene.

Por lo tanto, en esta situación, vamos a poder distinguir dos tipos de elementos:

- El **contenedor flex** que tendrá asignada la propiedad CSS *display:flex* y que va a controlar (ya veremos como) ciertas propiedades de los elementos que contiene. Los vemos en gris en la imagen inferior.
- Los **elementos flexibles** que son los elementos que están dentro del contenedor y cuyas propiedades modificaremos. Los vemos en azul en la imagen inferior.



Pero, ¿qué tipo de propiedades podremos modificar? Puede podremos modificar propiedades CSS como:

- La altura de los elementos flexibles.
- La anchura.
- El orden en el que se nos van a presentar.
- La alineación vertical.
- La alineación horizontal.
- La distribución de los elementos flexibles a lo largo del contenedor.

Es decir, vamos a poder controlar propiedades que usamos para maquetar y además, vamos a poder maquetar de manera mucho más ágil a lo que lo hacemos con las técnicas tradicionales de maquetado.

2 PROPIEDADES EN EL CONTENEDOR FLEX

El primer paso para maquetar usando elementos FLEX es añadir la propiedad FLEX al contenedor. En cuanto hagamos esto vamos a poder comprobar que empiezan a pasar cosas:

Si tengo este HTML:

```
<div class="container">
  <header>
    <div><p>Primera frase</p></div>
    <div><p>Segunda frase</p></div>
    <div><p>Tercera frase</p></div>
  </header>
</div>
```

Simplemente añadiendo la propiedad **display:flex** podremos ver que de manera automática los elementos ajustan su anchura a su contenido y flotan a la izquierda. Y todo es simplemente poniendo una sola regla CSS.

Además de con esta regla, desde el contenedor flex puedo modificar de manera directa varias propiedades de los elementos hijos. Por ejemplo:

- La dirección en la que se van a mostrar.
- Cómo se van a ajustar dentro del contenedor.
- La alineación horizontal de los elementos flexibles.
- La alineación vertical de los elementos flexibles cuando solo ocupan una línea.
- La alineación vertical de los elementos flexibles cuando ocupan más de una línea.

2.1 Dirección de los elementos flexibles.

La ajustaremos mediante la propiedad **flex-direction** que podrá tomar los siguientes valores:

- **row**: es la opción por defecto y ajustará los elementos flexibles de izquierda a derecha.
- **row-reverse**: igual que la anterior pero de derecha a izquierda.
- **column**: ajustará los elementos flexibles en columna, de arriba a abajo.
- **column-reverse**: igual que la de arriba pero de abajo a arriba.

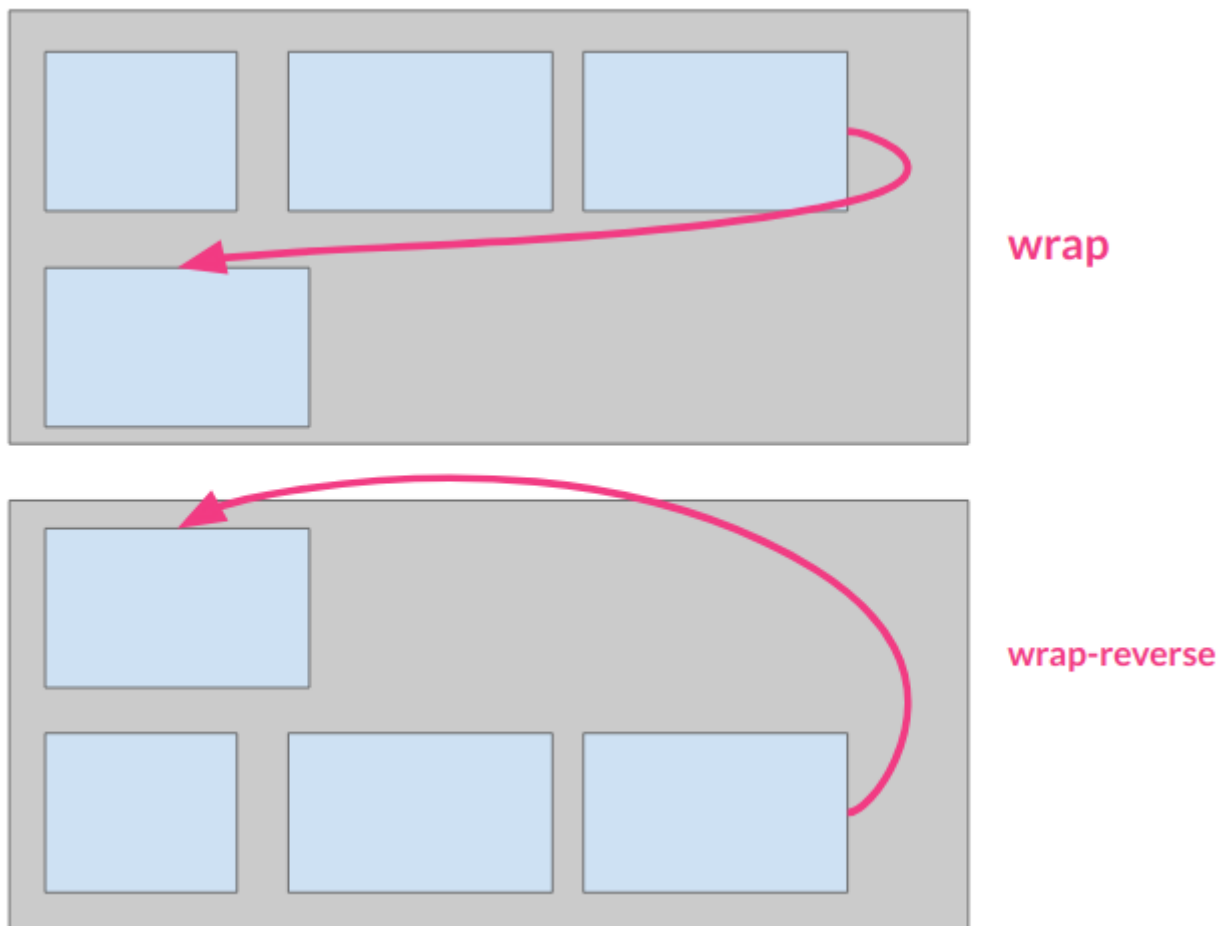
NOTA: Estamos dando por supuesto que trabajamos con sistemas de escritura occidentales donde el flujo de lectura es de izquierda a derecha y de arriba a abajo. En otros sistemas de escritura los valores por defecto en nuestro navegador podrían ser otros y deberíamos revisar todo con cuidado.

2.2 El ajuste de los elementos flexibles

Hemos visto como antes, por defecto y sin indicar ninguna anchura, los elementos flexibles adecuaban su tamaño a su contenido (si no les dábamos anchura) y se ponían todos a la izquierda permaneciendo siempre así aunque el ancho de la pantalla sea muy pequeño. Esto puede provocar desajustes en pantallas muy pequeñas. Este contratiempo podemos controlarlo usando la propiedad **flex-wrap** y eligiendo uno de los siguientes valores:

- **no-wrap**: es el valor por defecto y fuerza para que siempre los elementos estén en la misma línea aunque esto suponga que se salgan del contenedor (les haya dado o no les haya dado anchura).
- **wrap**: provoca un salto de línea si la anchura de los elementos (fijada por nosotros o por el contenedor) es superior a la del contenedor.
- **wrap-reverse**: lo mismo que arriba pero de abajo a arriba.

Podemos ver el efecto de los últimos valores en la siguiente imagen:



Las dos propiedades, **flex-direction** y **flex-wrap** podemos juntarlas en la propiedad **flex-flow** con dos partes como por ejemplo:

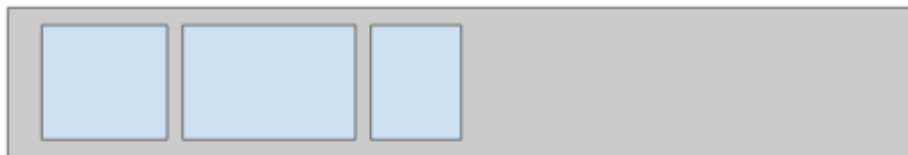
```
flex-flow: row wrap;
```

2.3 Alineación horizontal de los elementos flexibles

Podemos alinear horizontalmente los elementos flexibles, tengan o no tengan establecida una anchura, añadiendo la propiedad **justify-content** al elemento contenedor. Esta propiedad puede tener 6 valores distintos:

- **flex-start**: Los elementos flexibles se sitúan al principio.
- **flex-end**: Los elementos flexibles se situan al final.
- **center**: Los elementos se centran horizontalmente
- **space-between**: Distribuye el espacio restante entre los elementos pero el primero y el último están en los bordes.
- **space-around**: Distribuye el espacio restante entre los elementos pero no tiene en cuenta la distancia a los bordes.

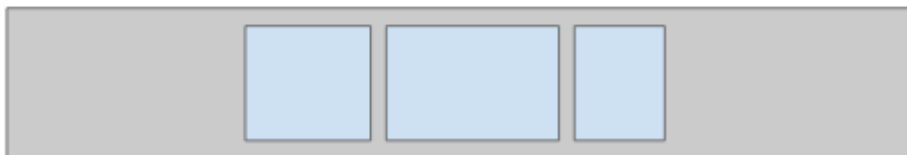
Visualmente:



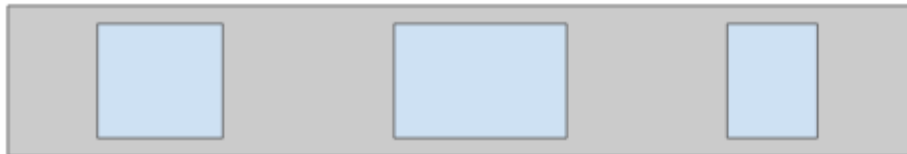
flex-start



flex-end



flex-center

**space-between****space-around**

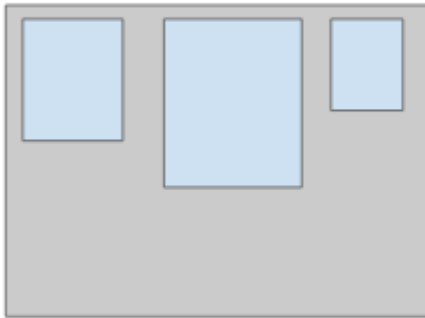
2.4 Alineación vertical de los elementos flexibles

Podemos alinear verticalmente los elementos flexibles añadiendo la propiedad **align-items** que puede tomar los siguientes valores:

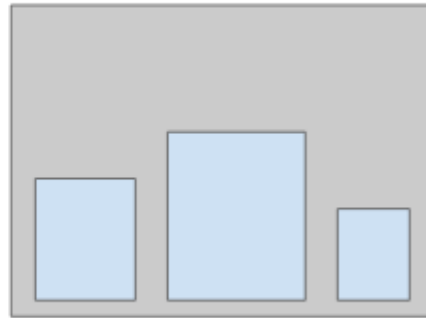
- **flex-start**: Los elementos se ponen junto al borde superior.
- **flex-end**: Los elementos se ponen junto al borde inferior.
- **center**: Los elementos flexibles se centran verticalmente.
- **stretch**: Los elementos crecen en altura para ocupar toda la altura del contenedor flexible. No deben tener altura fija establecida.
- **baseline**: Los elementos se alinean en relación con la primera línea de texto que posean los elementos flexibles.

Visualmente:

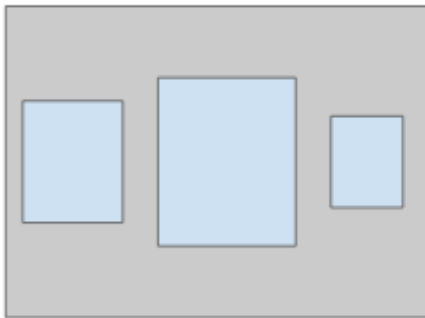
flex-start



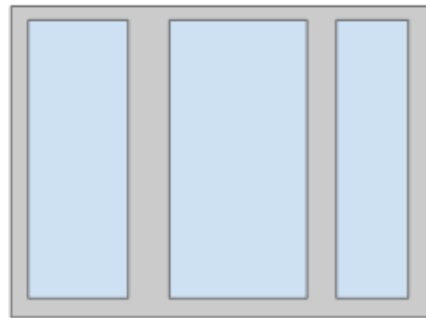
flex-end



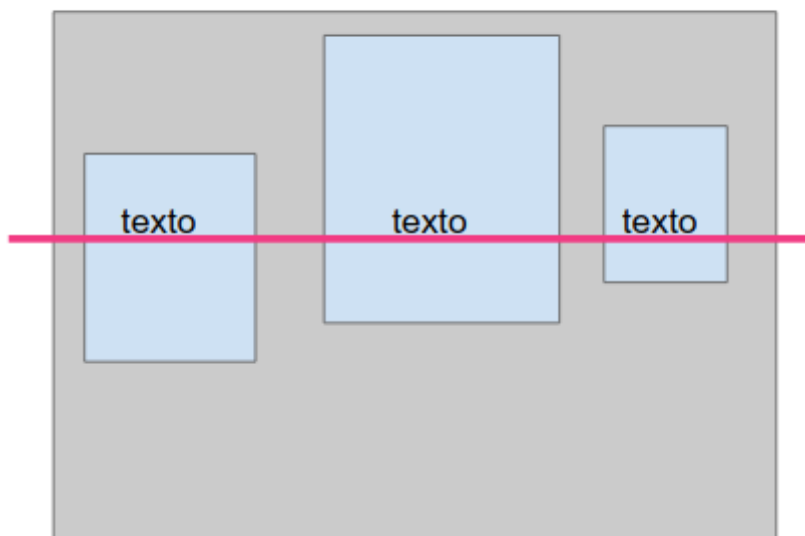
center



stretch (no height)



baseline



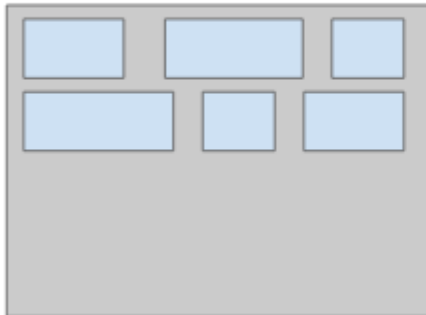
2.5 Alineación vertical - Wrap (cuando tengo varias líneas)

Si he usado la propiedad *flex-wrap:wrap* y resulta que tengo varias líneas de elementos flexibles también puedo alinearlas usando la propiedad CSS *align-content* con valores que son análogos a los vistos antes:

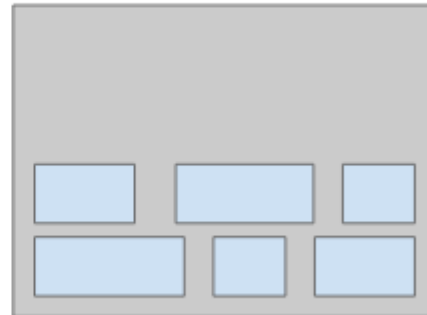
- flex-start
- flex-end
- center
- stretch
- space-between
- space-around

Visualmente:

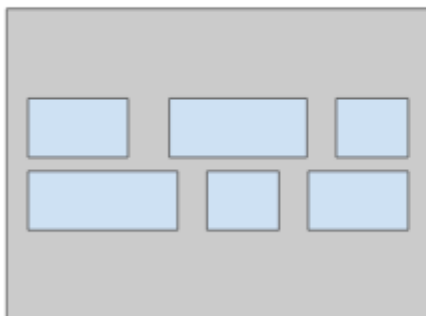
flex-start



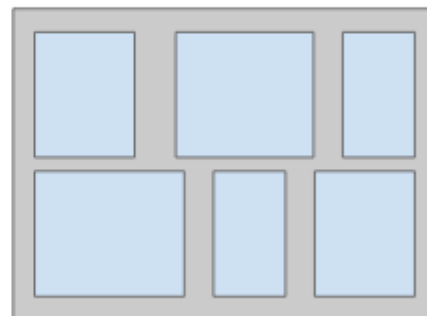
flex-end



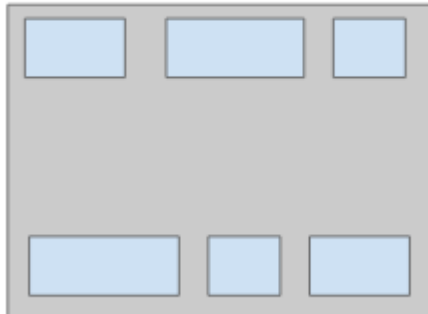
center



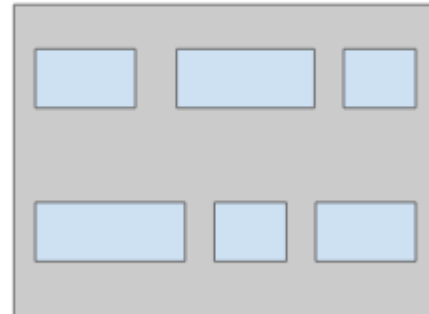
stretch (no height)



space-between



space-around



3 ELEMENTOS FLEXIBLES

Además de modificando las propiedades del contenedor FLEX, podemos actuar sobre los elementos flexibles modificando otra serie de propiedades relacionadas.

En este apartado nos vamos a ocupar de cómo modificar el orden en el que aparecerán los elementos flexibles y de cómo podemos controlar que se encojan o crezcan.

3.1 El orden de los elementos flexibles.

Los elementos flexibles se muestran dentro del contenedor flex en el mismo orden en que están escritos en nuestro código HTML.

Si queremos modificar esto debemos añadir la propiedad CSS **order** a los elementos cuyo orden queremos modificar.

Por defecto este valor es 0 y se mostrarán primeros aquellos elementos que tenga un mayor orden. En caso de empate se muestra antes el que primero estuviera en el código.

3.2 Ajustando el tamaño de los elementos flexibles.

Para controlar el tamaño de los elementos flexibles disponemos de varias propiedades relacionadas interesantes:

- **flex-grow:** que sirve para indicar, mediante un número, el factor de crecimiento de un elemento flexible cuando se distribuye entre los elementos flexibles el espacio restante. Por defecto es 1 pero si quiero que un elemento participe en el reparto debo añadirle esta propiedad.
- **flex-shrink:** que sirve para indicar, mediante un número el factor de contracción de un elemento flexible cuando el tamaño de todos sobrepasa el tamaño del contenedor. Por defecto es 1 pero si quiero que un elemento participe en la contracción debo añadirle esta propiedad.
- **flex-basis:** que sirve para indicar el tamaño de un elemento antes de que el espacio restante (negativo positivo) se distribuya. Por defecto el valor de esta propiedad es *auto* y hace que la anchura del elemento flexible se ajusta a su contenido.

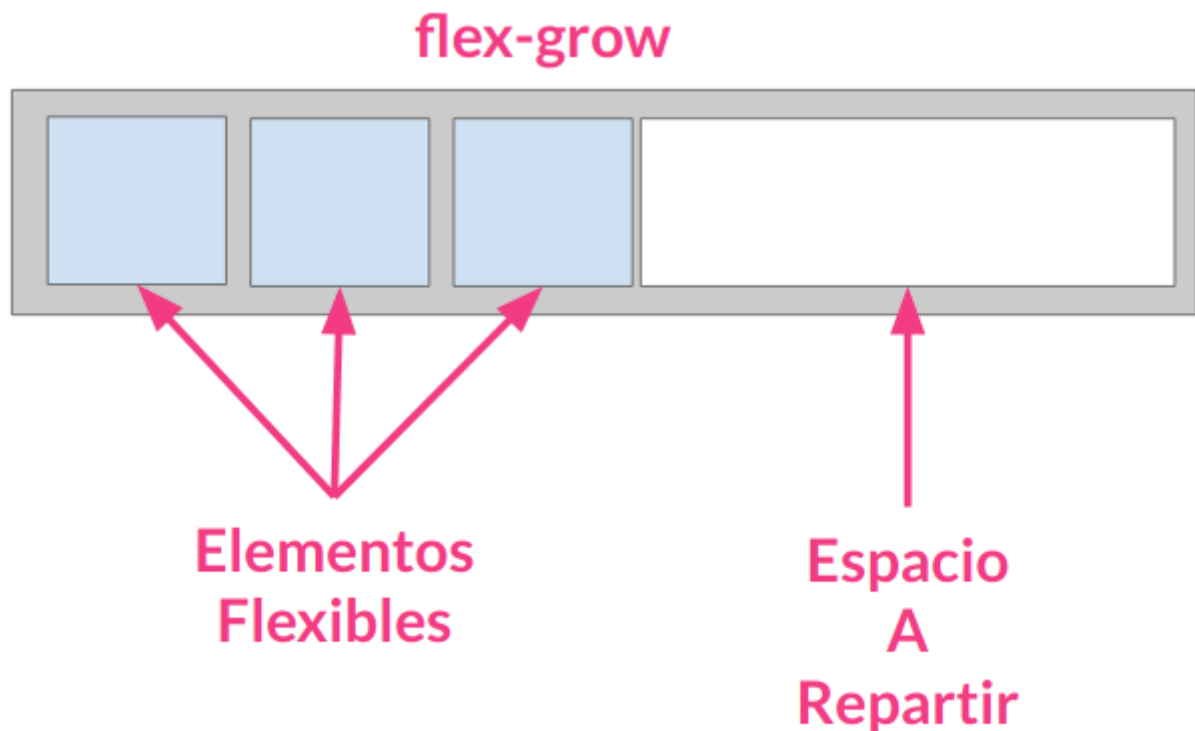
Debemos de tener en cuenta que para una correcta maquetación debemos considerar las

tres propiedades de manera conjunta. Además, se puede expresar de manera unitaria con la propiedad CSS **flex**. Por ejemplo:

```
flex: grow-factor shrink-factor flex-basis-value;
```

Para una total comprensión de cómo encogen/crecen estos elementos debemos entender qué es lo que se reparte.

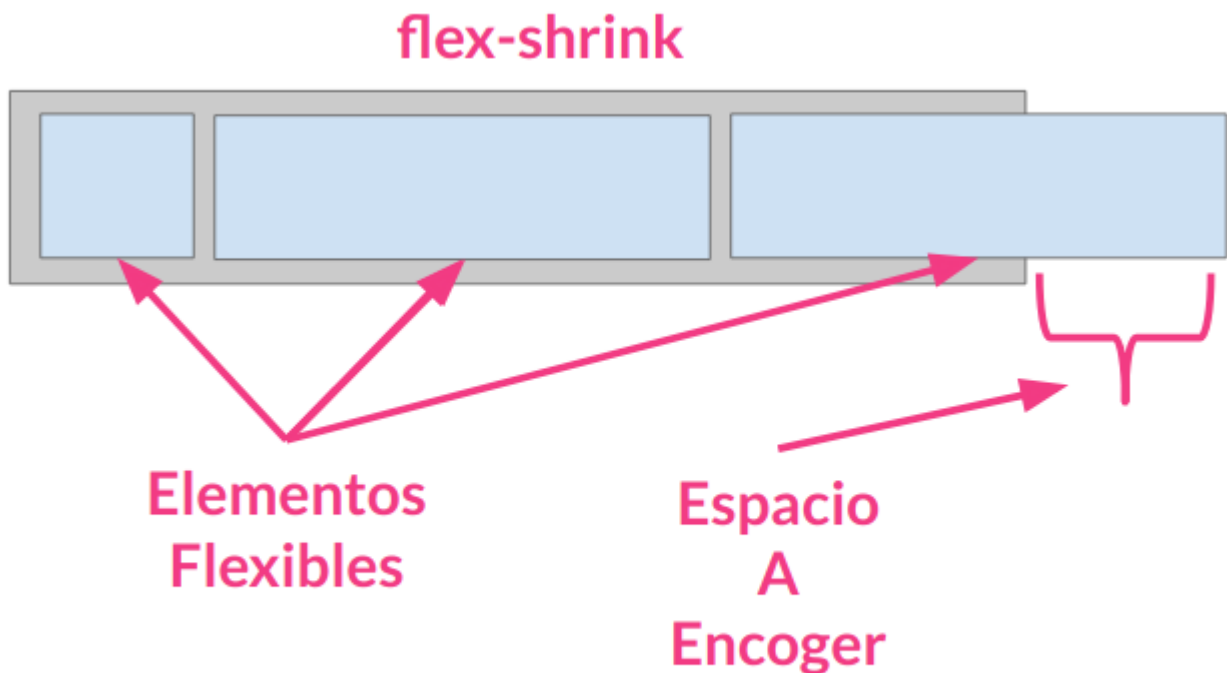
En caso de que los elementos deban crecer (*grow*):



Ese espacio a crecer se repartirá atendiendo al valor que tengan los elementos flexibles en la propiedad **flex-grow**.

Así si esos valores fueran: 2, 2 y 4 el espacio tendría 8 partes y cada uno crecería la cantidad de partes de su valor.

En caso de que los elementos deban encoger (*shrink*):



Ese espacio a encoger se repartirá atendiendo al valor que tengan los elementos flexibles en la propiedad **flex-shrink**.

Así si esos valores fueran: 2, 2 y 4 el espacio tendría 8 partes y cada uno encogería la cantidad de partes de su valor.

4 ELEMENTOS FLEXIBLES: ALINEACIÓN, JUSTIFICACIÓN Y ESPACIO LIBRE

En apartados anteriores hemos visto que desde el contenedor FLEX se puede establecer la alineación vertical de todos los elementos flexibles que contiene. Esto lo hacíamos con la propiedad CSS **align-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 mismos valores (y con el mismo significado) que la propiedad **align-items**.

- flex-start
- flex-end
- center
- stretch (no debe tener altura establecida)
- baseline

NOTA: Los elementos flex no hace caso a la propiedad **vertical-align**. Podemos ver un ejemplo de todo esto en la siguiente imagen:

