

DISPLAY FLEX

Cajas Flexibles

El nuevo display: CAJA FLEXIBLE

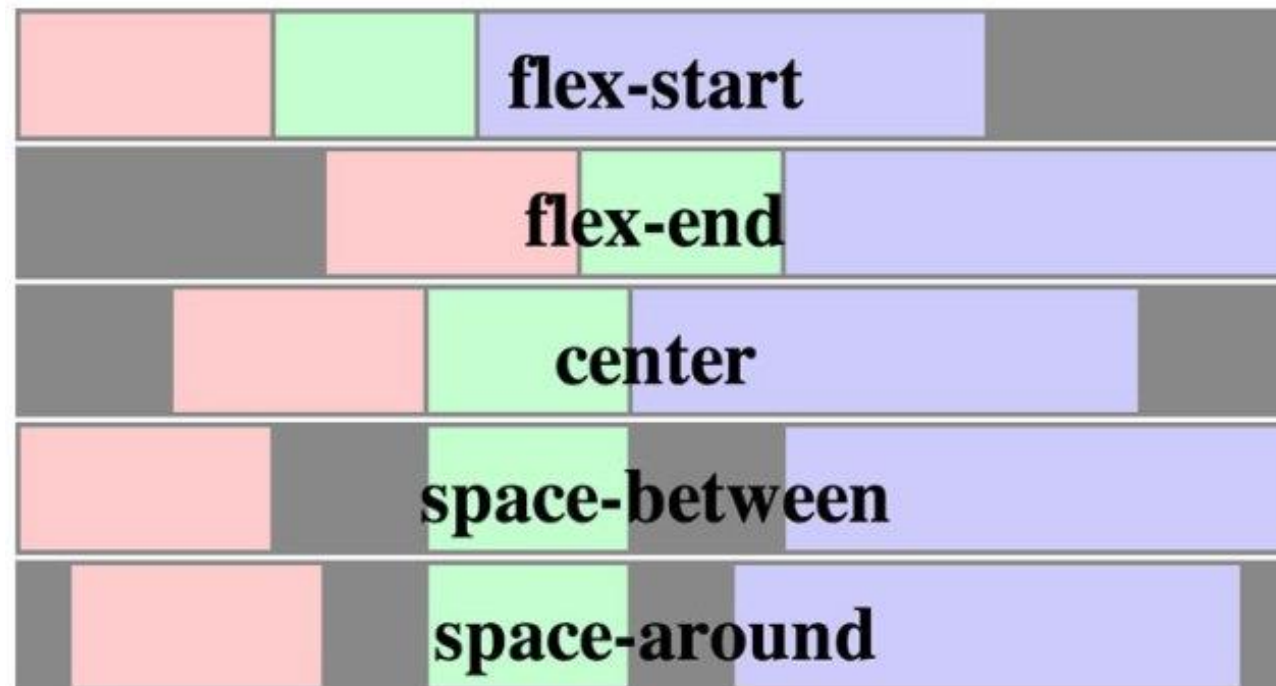
- Podemos generar **contenedores flexibles** con `display: flex` o `inline-flex`) Permite manipular el flujo de los elementos de bloque que contiene llamados **flex items**.
- **Sus hijos** modifican su tamaño para llenar el espacio disponible, distribuir el espacio entre ellos y alterar el orden de maquetación.
- Permite establecer el eje X o el Y como principal y el sentido del flujo de los items en ambos ejes.

El nuevo display: CAJA FLEXIBLE

- El contenedor define **2 ejes**: el **principal** (main axis), en donde los items se ordenan, y el **transversal** (cross axis) perpendicular al anterior.
- **Orientación de los items en el eje principal**
 - Dirección `flex-direction: row, row-reverse, column, column-reverse`
 - Cantidad de líneas `flex-wrap: no-wrap, wrap, wrap-reverse`
 - Propiedad shorthand `flex-flow: row wrap`

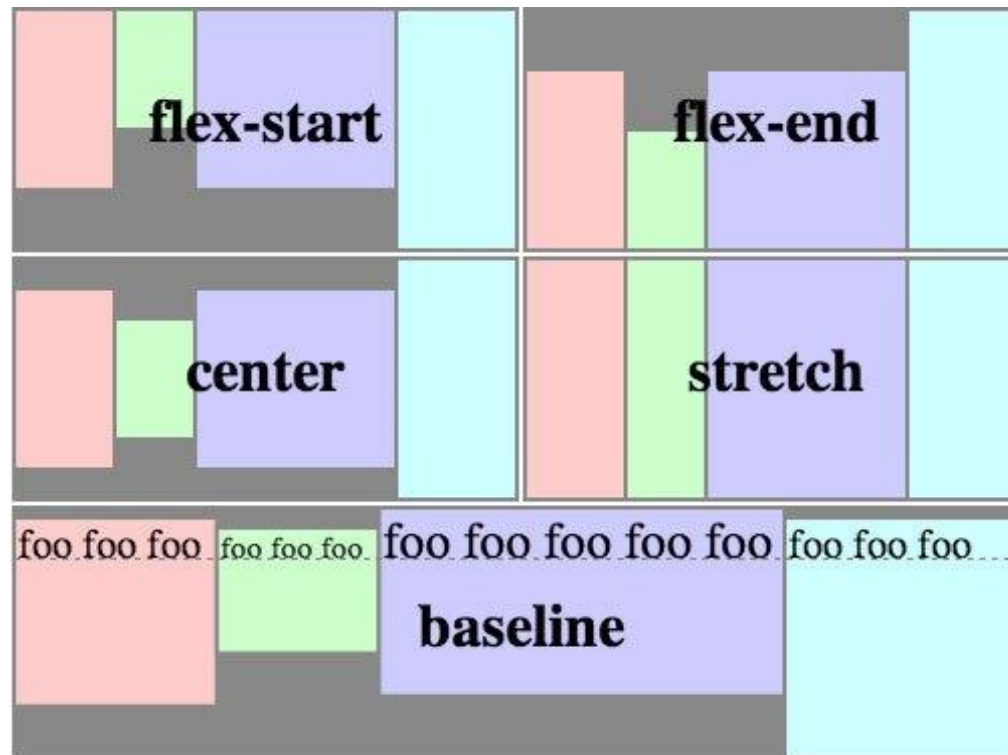
El nuevo display CAJA FLEXIBLE

- **Alinear en eje principal** (no tiene efecto en cajas con una sola línea) `justify-content: flex-start, flex-end, center, space-between, space-around`



El nuevo display CAJA FLEXIBLE

- **Alinear en eje transversal:** `align-items`: `flex-start`, `flex-end`, `center`, `baseline`, `stretch`



Organización de los hijos

- **flex** tamaño de cada hijo en relación a los demás (proporción), **1**: todos iguales, **2**: el doble que el resto... Acepta **3 parámetros**:
 - ▣ **flex-grow**: cuanto crece en relación al resto (default 0).
 - ▣ **flex-shrink**: cuanto se reduce en relación al resto (default 1).
 - ▣ **flex-basis**: tamaño inicial antes de crecer o reducirse.
- **order** determina el orden en que se muestran, **0** es el del html, se puede cambiar dándole un valor de orden diferente a cada hijo, **-1** por encima del primero.
- **align-self**: alinea el hijo (a si mismo) con respecto al eje transversal (pisa el valor de align-items).

Ejemplo tamaño hijos

- Suponiendo que tenemos un contenedor flex de 300 px de ancho y 2 hijos con las siguientes valores de ***flex-grow***, ***flex-shrink*** y ***flex-basis***:
 - ▣ #uno{ flex: 3 1 100px; }
 - ▣ #dos{ flex: 1 2 100px; }
 - ▣ Los 100px que faltan se reparten 3:1 por flex-grow.
- Si ahora el tamaño del contenedor es 170px (con los mismos valores de flex de los hijos):
 - ▣ Faltan 30px en el contenedor.
 - ▣ Se reducen 1:2 por el valor de flex-shrink.