FlexBox

**🧩 1. Contenedor de tipo flex**

* Se define un contenedor flexible con display: flex o display: inline-flex.
* Los elementos hijos dentro de este contenedor se convierten automáticamente en **elementos flexibles**.
* flex se comporta como un elemento de bloque, mientras que inline-flex mantiene el contenedor en línea.

**🔀 2. Ejes de Flexbox**

* **Eje principal**: definido por flex-direction.
* **Eje cruzado**: perpendicular al principal.
* Todo el comportamiento de alineación y ordenamiento en flexbox depende de estos ejes.

**flex-direction**

* Define la dirección del flujo de los elementos:
  + row (horizontal izquierda→derecha)
  + row-reverse (horizontal derecha→izquierda)
  + column (vertical arriba→abajo)
  + column-reverse (vertical abajo→arriba)

**↩️ 3. Flex-wrap**

* Controla si los elementos se mantienen en una sola línea o pueden “saltar” a la siguiente.
  + nowrap (por defecto): todos en una sola línea.
  + wrap: los elementos se ajustan en varias líneas cuando no caben.

**🔢 4. Order**

* Permite **cambiar el orden visual** de los elementos, independientemente del orden en el HTML.
* Útil para diseños adaptativos (responsive), donde el orden cambia según el dispositivo.

**🔄 5. Flex-flow**

* Propiedad **shorthand** que combina:
  + flex-direction
  + flex-wrap  
    Ejemplo: flex-flow: row wrap;

**⚖️ 6. Flex-grow, flex-shrink y flex-basis**

Propiedades que determinan **cómo los elementos flexibles crecen o se reducen** dentro del contenedor.

**flex-grow**

* Controla **cómo crecen los elementos** cuando hay espacio disponible.
* Valor por defecto: 0 (no crece).
* Cuanto mayor sea el valor, más espacio adicional ocupará el elemento.
* Usos comunes:
  + Barras de navegación y menús.
  + Diseños homogéneos de tarjetas.
  + Paneles ajustables.
  + Formularios responsivos.
  + Footers distribuidos.
  + Galerías dinámicas.

**flex-shrink**

* Controla **cómo se encogen los elementos** cuando no hay suficiente espacio.
* Valor por defecto: 1 (se encoge proporcionalmente).
* Usos:
  + Evitar desbordamiento.
  + Priorizar contenido importante.
  + Mantener cohesión del diseño.
  + Balancear espacio entre contenido y estructura.

**flex-basis**

* Define el **tamaño base inicial** del elemento antes de distribuir el espacio.
* Puede sobrescribir width o height según el eje principal.
* Valores:
  + auto: usa el tamaño del contenido (por defecto).
  + 0: el tamaño base es el mínimo posible.
* Afecta el cálculo conjunto con flex-grow y flex-shrink.

**Propiedad shorthand flex**

* Combina las tres propiedades anteriores:
* flex: [grow] [shrink] [basis];
* Valor por defecto: 0 1 auto.
* Ejemplo:
  + flex: 1; → crece y se encoge proporcionalmente, tamaño base 0.
  + flex: 1 0 200px; → crece si hay espacio, tamaño base 200px.

**🧭 7. Alineación de elementos**

Propiedades usadas para alinear los hijos dentro del contenedor flexible.

**justify-content**

* Alinea los elementos **a lo largo del eje principal**.
* Valores:  
  flex-start, flex-end, center, space-between, space-around, space-evenly.

**align-items**

* Alinea los elementos **en el eje cruzado** (por defecto, vertical).
* Valores:  
  flex-start, flex-end, center, stretch, baseline.

**align-content**

* Alinea **varias líneas** de elementos flexibles en el eje cruzado (solo si hay más de una línea).
* Valores:  
  flex-start, flex-end, center, space-between, space-around, stretch.

**align-self**

* Alinea un **elemento individual** de forma diferente al resto.
* Sobrescribe el valor de align-items para ese ítem concreto.

**📘 8. Resumen general**

* Flexbox necesita un contenedor padre con display: flex.
* Dispone de dos ejes (principal y cruzado).
* Controla crecimiento (flex-grow), contracción (flex-shrink), y tamaño base (flex-basis).
* Permite reordenar (order) y alinear (justify-content, align-items, align-content, align-self) los elementos.
* Se puede simplificar la configuración con flex-flow y flex.