

# Flexbox

## Hojas de Estilo en Cascada

# Flexbox

Ah, Flexbox.

También conocido como “finalmente puedo acomodar las cosas”, “CSS no está tan mal”,

y “wow, hoy no quise lanzar mi monitor por la ventana” 😊

# Flexbox

Flexbox (o Flexible Box Layout Module) es un modelo de diseño muy útil para crear layouts flexibles y responsivos, tanto en filas como en columnas.

Un elemento se convierte en un contenedor flexible al aplicar:

```
display: flex;
```

Todos los elementos hijos se vuelven ítems flexibles (flex items), y pueden distribuirse y alinearse de forma automática según las propiedades que se apliquen.

# Flexbox

## Propiedades principales del contenedor (padre)

flex-direction

Define la dirección principal del eje.

Valores:

row → en fila (por defecto)

row-reverse → fila inversa

column → en columna

column-reverse → columna inversa

# Flexbox

justify-content

Alinea los elementos a lo largo del eje principal.

Valores:

flex-start, flex-end, center,

space-around, space-between, space-evenly

# Flexbox

align-items

Alinea los elementos a lo largo del eje cruzado (el eje perpendicular).

Valores:

flex-start, flex-end, center, baseline, stretch

flex-wrap

Permite que los elementos se ajusten a nuevas líneas si no caben.

Valores:

nowrap, wrap, wrap-reverse

# Flexbox

## align-content

Controla la alineación de varias líneas dentro del contenedor.

Valores:

flex-start, flex-end, center,

space-around, space-between, stretch

# Flexbox

flex-flow

Atajo que combina flex-direction y flex-wrap.

flex-flow: row wrap;

gap

Define el espacio entre los ítems flexibles.

También puede usarse row-gap o column-gap.

# Flexbox

## Propiedades del ítem (hijo)

align-self

Permite sobrescribir la alineación del align-items solo para ese elemento.

flex-basis

Define el tamaño inicial del ítem (ancho o alto dependiendo de la dirección del eje).

flex-grow

Indica cuánto puede crecer un ítem si hay espacio adicional.

Ejemplo: si un ítem tiene flex-grow: 2, crecerá el doble que uno con flex-grow: 1.

# Flexbox

## flex-shrink

Controla si un ítem puede reducir su tamaño si no hay espacio suficiente.

Un valor de 0 evita que se encoja.

# Flexbox

flex

Propiedad abreviada que combina:

flex: flex-grow flex-shrink flex-basis;

# Flexbox

## order

Permite cambiar el orden visual de los elementos sin modificar el HTML.

Por defecto, todos tienen order: 0.

# Flexbox

Concepto	Qué hace
`display: flex;`	Activa el modo flexbox
`flex-direction`	Define la dirección principal (fila o columna)
`justify-content`	Alinea los ítems en el eje principal
`align-items`	Alinea en el eje perpendicular
`flex-wrap`	Permite que los ítems salten a otra línea
`gap`	Espaciado entre ítems
`order`	Cambia el orden visual
`flex-grow / shrink`	Controlan crecimiento o reducción

# Flexbox

## Flexbox – Ejemplos visuales

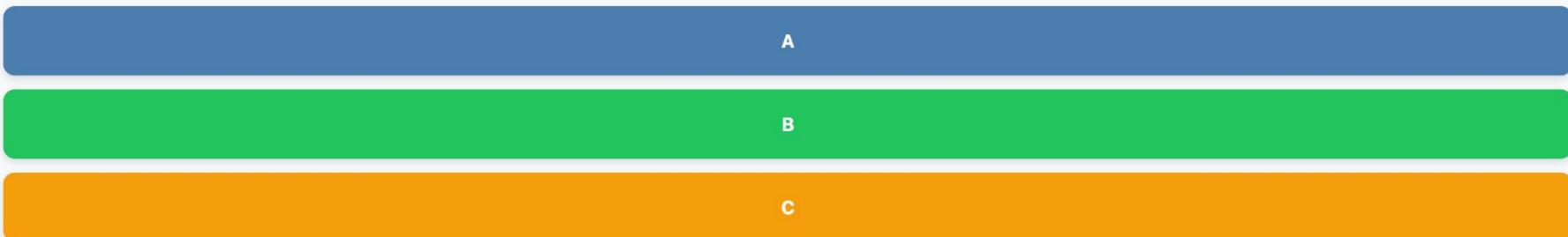
Edita las clases en el HTML/CSS para observar cada comportamiento.

### 1) flex-direction

Contenedor: `display:flex; flex-direction: row;`



Contenedor: `display:flex; flex-direction: column;`



# Flexbox

Esta propiedad define la dirección principal en la que los elementos flexibles se organizan dentro del contenedor.

Por defecto, los elementos se colocan en fila (row), de izquierda a derecha.

También se puede cambiar a columna (column), o invertir el orden con row-reverse o column-reverse.

Explicación visual:

En la primera parte del ejemplo, se ve cómo los elementos cambian su orientación:

En la fila, se alinean horizontalmente uno al lado del otro.

En la columna, se apilan verticalmente.

# Flexbox

## 2) justify-content (eje principal)

```
justify-content: flex-start;
```



```
justify-content: center;
```



```
justify-content: space-between;
```



```
justify-content: space-around;
```



```
justify-content: space-evenly;
```



# Flexbox

Controla cómo los elementos se distribuyen a lo largo del eje principal (horizontal si usamos row, vertical si usamos column).

Valores más comunes:

flex-start: todos al inicio.

center: todos centrados.

flex-end: todos al final.

space-between: espacio igual entre los ítems, sin margen en los extremos.

space-around: espacio igual alrededor de cada ítem.

space-evenly: espacio totalmente uniforme entre y alrededor de los ítems.

# Flexbox

En esta parte se observa cómo el espacio cambia entre las cajas dependiendo del valor.

Esto permite lograr alineaciones precisas y distribuciones equilibradas sin usar márgenes manuales.

# Flexbox

## 3) align-items (eje cruzado)

align-items: flex-start; (cajas con alturas variadas)



align-items: center;



align-items: baseline;



# Flexbox

Define cómo se alinean los elementos en el eje cruzado (el eje perpendicular a flex-direction).

Valores más comunes:

flex-start → arriba o a la izquierda

center → centrados

flex-end → abajo o a la derecha

baseline → alinea según la línea base del texto

stretch → los estira para llenar el contenedor

# Flexbox

En el ejemplo se usaron cajas con alturas diferentes para que se vea el cambio al modificar align-items.

Se nota cómo al usar center las cajas se alinean perfectamente en medio del contenedor, mientras que con baseline se alinean por el texto.

# Flexbox

## 4) flex-wrap y gap

flex-wrap: nowrap; gap: 12px;



flex-wrap: wrap; gap: 12px; (los ítems saltan a otra línea)



# Flexbox

Descripción:

flex-wrap indica si los elementos pueden ajustarse a otra línea cuando no hay espacio suficiente.

nowrap → todos en una sola línea.

wrap → saltan a una nueva línea cuando es necesario.

wrap-reverse → igual que wrap, pero en sentido inverso.

gap define el espacio entre los elementos, sin usar márgenes individuales.

# Flexbox

En las capturas puedes mostrar cómo al usar wrap las cajas se acomodan automáticamente a la siguiente fila sin deformar el diseño.

El gap mantiene el mismo espacio entre todas las cajas, de forma ordenada y consistente.

# Flexbox

## 5) Propiedades de ítems

```
.a {flex: 1 1 150px}, .b {flex: 2 1 150px}, .c {flex: 0 0 150px}
```

grow:1

grow:2

shrink:0

order y align-self (solo este ítem cambia)

B (order:-1 · align-self:end)

A

C

# Flexbox

Estas propiedades se aplican a los elementos hijos dentro del contenedor flexible y controlan su tamaño, crecimiento, y posición.

Principales:

flex-grow: permite que un ítem crezca cuando hay espacio libre.

flex-shrink: define si el ítem se encoge cuando hay poco espacio.

flex-basis: tamaño base inicial.

order: cambia el orden visual sin alterar el HTML.

align-self: sobrescribe la alineación solo para ese elemento.

# Flexbox

En la demo se muestra cómo una caja puede ocupar más espacio (`flex-grow:2`) o quedarse fija (`flex-shrink:0`).

Además, se ve cómo `order:-1` puede mover un elemento al inicio, sin cambiar su posición en el código HTML.

Código disponible en

<https://github.com/LuisContrerasGlz/ClaseFundamentosDeProgramacion>