

Esta clase va a ser

- grabada

Certificados oficialmente por



CODERHOUSE

Clase 05. DESARROLLO WEB

Flexbox

Certificados oficialmente por



CODERHOUSE

Glosario

Camel Case (Joroba de camello): es un estilo de escritura que permite que se puedan leer de forma más simple palabras compuestas.

Box model: el concepto de que “todo es una caja”, da lugar al modelo de posicionamiento box model. Sin importar si son de línea o de bloque (pero tienen su incidencia en lo que sean), todas las etiquetas tienen propiedades en común.

Width: es la propiedad CSS que controla la anchura de la caja de los elementos.

Height: es la propiedad CSS que controla la altura de la caja de los elementos.

Overflow: es una propiedad que tiene 4 valores posibles. Ellos son: visible, hidden, scroll y auto.

Glosario

Padding (relleno): las propiedades padding-top, padding-right, padding-bottom y padding-left, se utilizan para definir los espacios internos de cada uno de los lados del elemento, por separado.

Border: las propiedades border-top, border-right, border-bottom, y border-left, se utilizan para definir los bordes de cada lado del elemento por separado.

Margin (márgenes): las propiedades margin-top, margin-right, margin-bottom y margin-left, se utilizan para definir los márgenes de cada uno de los lados del elemento por separado.

Objetivos de la clase

- **Conocer** el modelo flexbox de disposición de elementos.
- **Aplicar** sus diferentes variantes.
- **Adquirir** los conocimientos necesarios para comprender y crear una página con Flexbox

Temario

04

CSS + Box Modeling

- ✓ Tipografías
- ✓ Box Modeling
- ✓ Display
- ✓ Posiciones

05

Flexbox

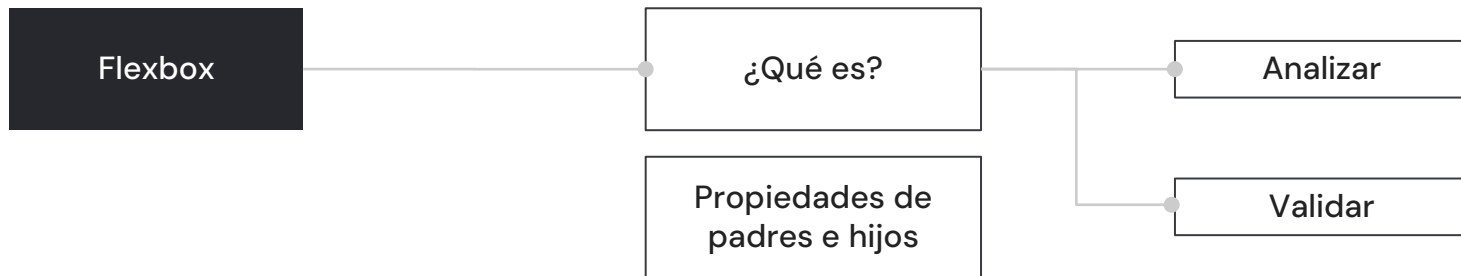
- ✓ [Flexbox](#)
- ✓ [Propiedad de padres e hijos](#)

06

Grids

- ✓ Grids
- ✓ Grids y Flexbox
- ✓ Grids por áreas

MAPA DE CONCEPTOS



Flexbox

¿Qué es flexbox?

Definición

Flexbox es un modo de diseño que nos permite crear estructuras para sitios web de una forma más fácil que con positions.

Con Flexbox podrás posicionar y distribuir los elementos como tú quieras.



Características

- ✓ Es un conjunto de propiedades de CSS.
- ✓ Se basa en un contenedor (padre) para ordenar a sus ítems (hijos).
- ✓ Nos permite posicionar elementos vertical y horizontalmente, establecer cómo se distribuirán, el orden que tendrán e incluso el tamaño en proporción a otros elementos.
- ✓ Esto es perfecto para crear diseños adaptables a dispositivos móviles (Responsive Design).

Beneficios

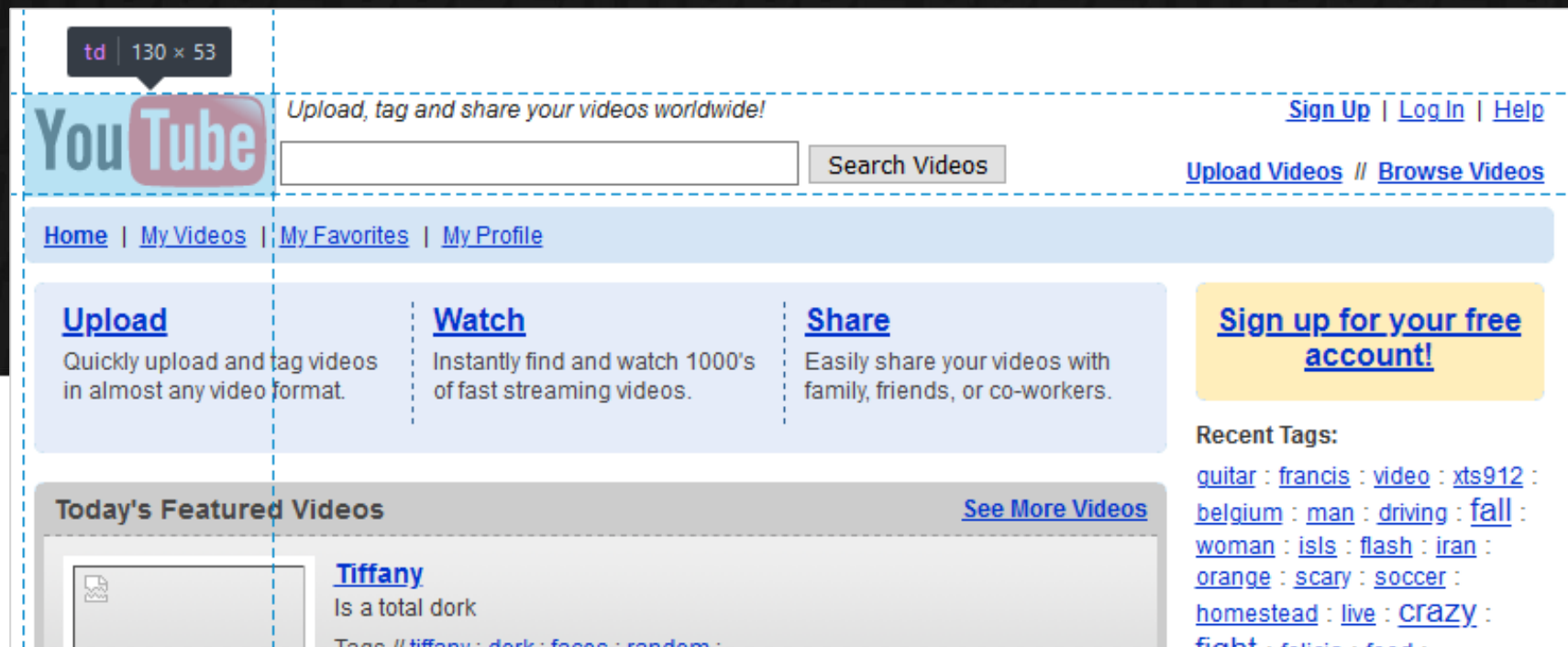
- ✓ Distribuir los elementos en sentido vertical y horizontal.
- ✓ Facilitar la adaptación del contenido en distintos dispositivos.
- ✓ Redefinir el sentido del flujo de los elementos (hacia arriba, hacia abajo, hacia la izquierda o hacia la derecha).
- ✓ Alinear los elementos respecto al padre o respecto a sus hermanos.

Algo de historia

Veamos, brevemente, un poco de historia acerca del modo en que ha evolucionado la forma de crear estructuras...



Antes se usaban mucho las tablas



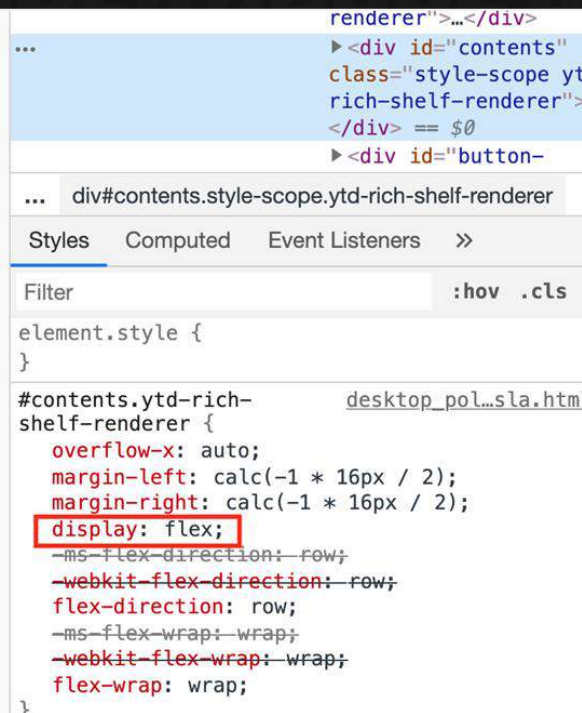
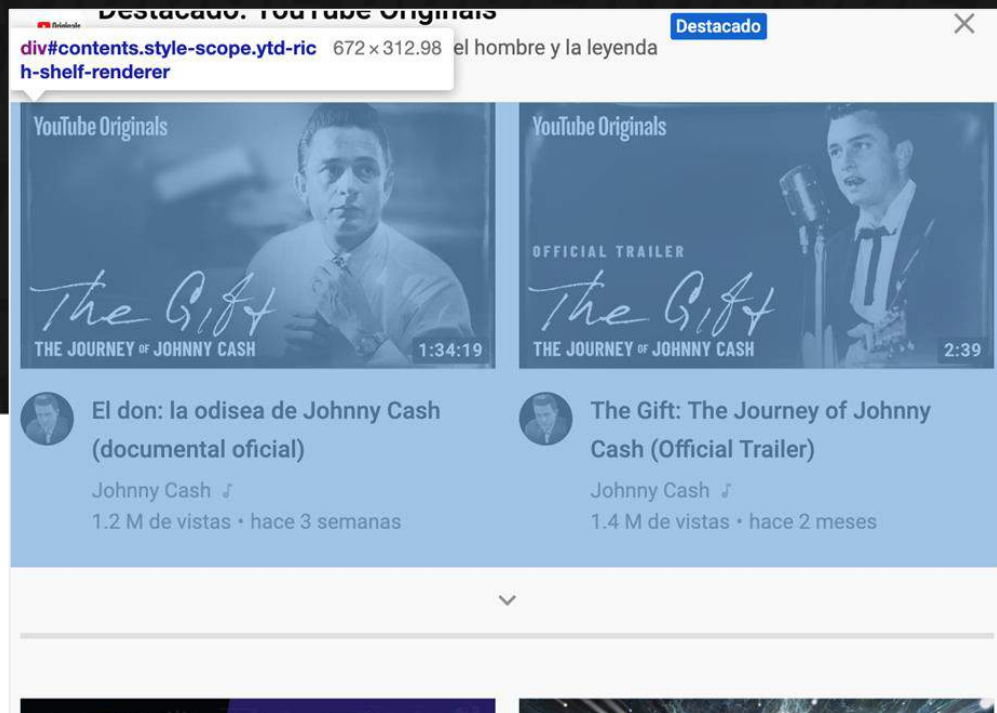
Algún tiempo atrás (no tanto)... se usaba el elemento Float

The image is a screenshot of a web browser displaying the YouTube homepage from 2010. The browser's address bar shows the URL `#homepage-main-content.homepage-non-interactive.homepage-new-skin` and the dimensions `625 x 1507`. The page content includes a header with the text "Interview with Obama" and a main section with four video thumbnails. Each video has a title, a timestamp, and a description. The first video is titled "Why are banks reluctant to modif..." and the second is "State of the Union Question". The third video is titled "President Obama - How will you h...". The fourth video is titled "President Obama - How will you h...". The page also features a sidebar with sections like "What's New", "New Product Id", "YouTube Badge", and "Your Questions". On the right side, there is a CSS inspector showing the styles for the element `#homepage-main-con`. The styles include `float: left;`, `clear: left;`, `width: 640px;`, and `margin-top: 5px`. The `float: left;` property is highlighted with a red box.

#homepage-main-content {
float: left;
clear: left;
width: 640px;
margin-top: 5px
}

Fuente: <https://web.archive.org/web/20100202095628/http://www.youtube.com/>

En la actualidad se aplica Flexbox





Ejemplo en vivo

¡Vamos a practicar lo visto!

Profe, comparta en su pantalla una página web que aplique **Flexbox** 



PARA RECORDAR

Recuerda

Para poder visualizar los estilos CSS aplicados a una página (incluidas las propiedades de flexbox), utilizamos el Inspector de nuestro navegador. Para acceder al mismo, debemos hacer **click derecho → inspeccionar, o utilizar la tecla F12.**

Propiedades de padres e hijos

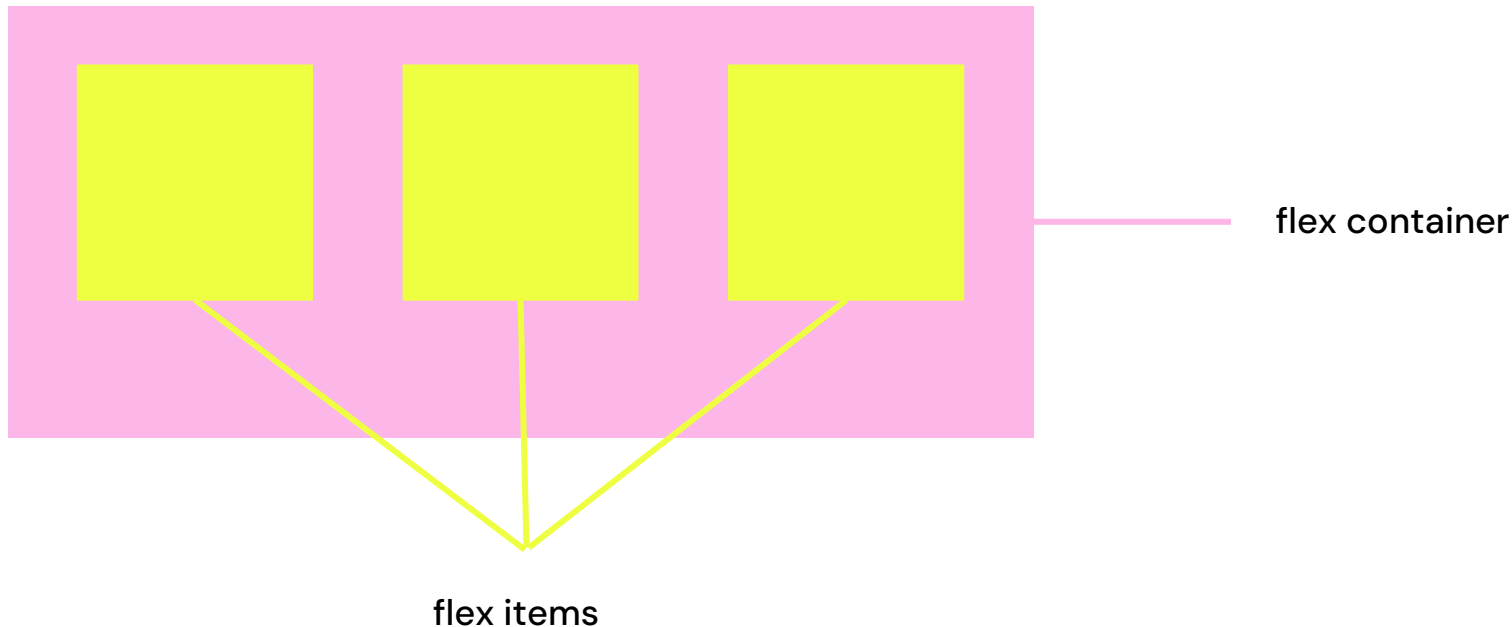
Propiedades para aplicar en el contenedor flexible (el padre)

Display	Flex (el que inicia): indicará que sus hijos serán “flexibles”.
flex-direction	Elegir dirección vertical u horizontal.
flex-wrap	¿Se hará multilínea cuando llegue al límite?
flex-flow	Abreviación de propiedades.

Propiedades para aplicar en el contenedor flexible (el padre)

justify-content	Alinear horizontalmente a los hijos si el padre es "fila" o verticalmente si el padre es "columna".
align-items	Alinear verticalmente a los hijos (si el padre es "columna").
align-content	Alinear verticalmente a los hijos cuando son multilínea.

Propiedades del padre



¿Cómo empezamos?

Lo primero que debemos hacer es establecer la propiedad display con el valor flex en el elemento padre.

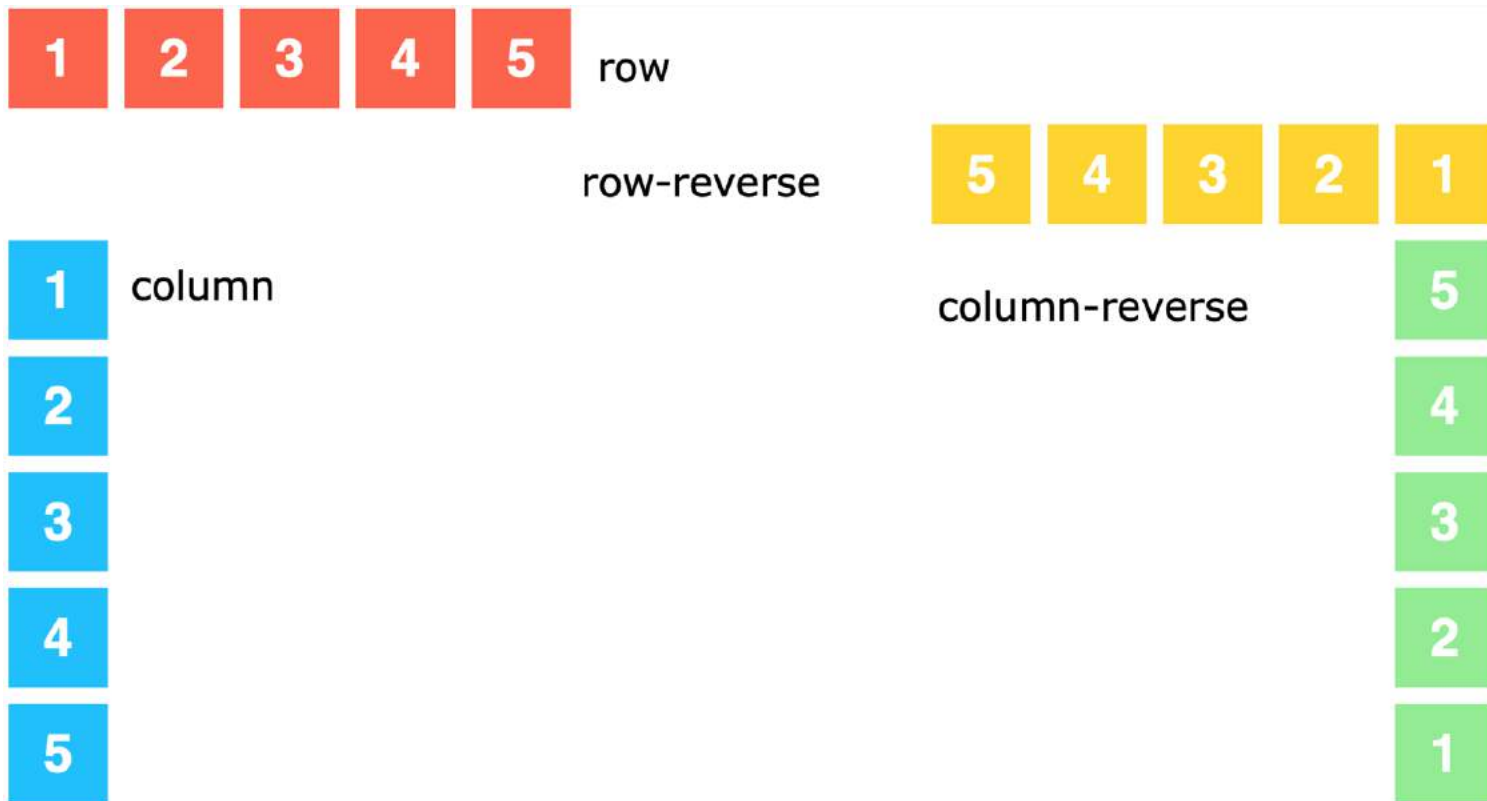
```
.padre-flex {  
  display: flex;  
}
```

Flex-direction

Esta propiedad nos va a permitir especificar si queremos que los flex items se dispongan en filas o columnas. Sus valores posibles son:

- row
- row-reverse
- column
- column-reverse

```
.padre-flex {  
    display: flex;  
    flex-direction: row;  
    /* predeterminado */  
}
```

Resumen de Flex-direction

Flex-wrap

El comportamiento inicial del contenedor flexible es poder mantener los flex items en su eje, sin importar que las dimensiones de los mismos cambien.

Con flex-wrap vamos a poder especificar si queremos que los ítems puedan saltar a una nueva línea cuando el contenedor flexible se quede sin espacio.

Sus valores posibles son:

- nowrap
- wrap
- wrap-reverse

```
.padre-flex {  
    display: flex;  
    flex-wrap: no-wrap;  
    /* predeterminado */  
}
```



nowrap



wrap



wrap-reverse

Resumen de Flex-wrap

Flex-flow

Es la forma abreviada (shorthand) o rápida para las propiedades:

- ✓ flex-direction
- ✓ flex-wrap

Se pone primero el valor de flex-direction, y luego el de flex-wrap.

```
.padre-flex {  
    display: flex;  
    flex-flow: row nowrap;  
}
```



Break

¡10 minutos y volvemos!

¿Preguntas?

#CoderTip: Ingresa al [siguiente link](#) y revisa el material interactivo que preparamos sobre **Preguntas Frecuentes**, estamos seguros de que allí encontrarás algunas respuestas.

Justify-content

Justify-content nos va a permitir **alinear los elementos en el eje principal**. Este puede ser el eje vertical u horizontal, según lo especifiquemos con flex-direction. Con esta propiedad podemos distribuir los flex items (hijos) en el contenedor (padre), cuando los ítems no utilicen todo el espacio disponible en su eje principal actual.

Sus valores posibles son:

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

```
.padre-flex {  
    display: flex;  
    flex-direction: row;  
    justify-content: flex-start;  
    /* predeterminado */  
}
```

1 2 3 4 5 flex-start

flex-end 1 2 3 4 5

1 2 3 4 5 flex-center

1 2 space-between 3 4 5

1 2 space-around 3 4 5

1 2 space-evenly 3 4 5

Justify-content resumen

Align-items

Así como justify-content alinea los elementos en el eje principal, align-items hace lo propio, pero en el eje secundario. Este será el opuesto al principal

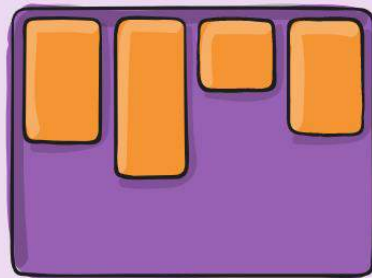
Sus valores posibles son:

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

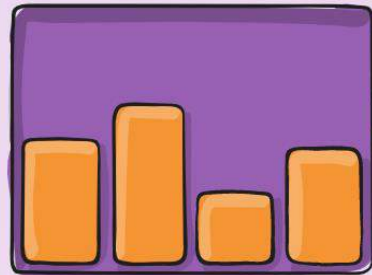
```
.padre-flex {  
    display: flex;  
    flex-direction: row;  
    align-items: flex-start;  
    /* predeterminado */  
}
```

Align-items resumen

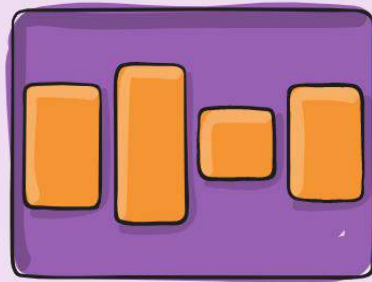
flex-start



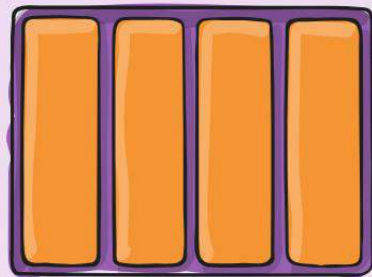
flex-end



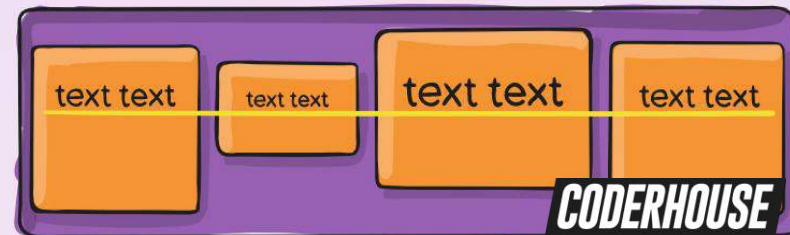
center



stretch



baseline



Align-content

Esta propiedad solo tiene efecto cuando el contenedor flexible tiene varias líneas de flex items (hijos). Si se colocan en una sola línea, esta propiedad no tiene ningún efecto sobre el diseño.

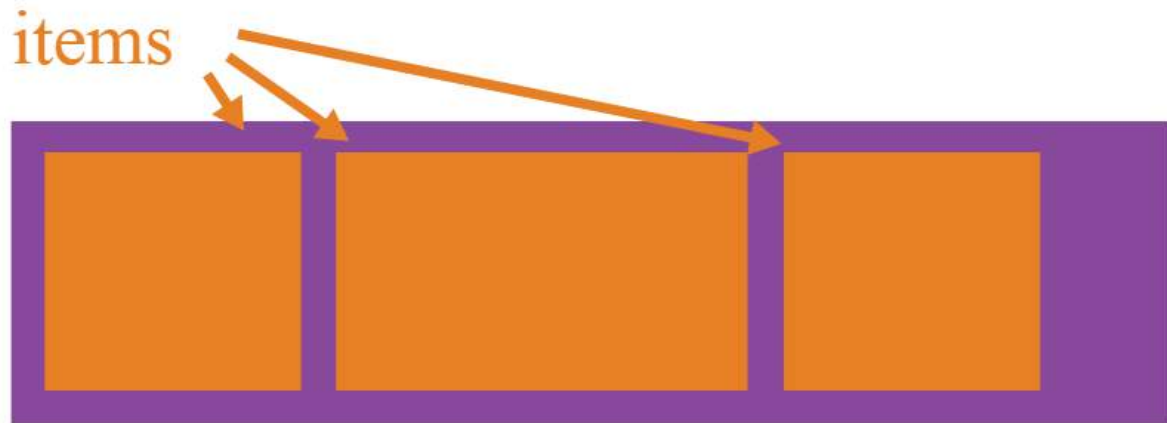
Para poder aplicarlo se necesita tener el atributo `flex-wrap`, que permita verificar los ejes horizontales.

Su valores posibles son:

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

```
.padre-flex {  
    display: flex;  
    flex-wrap: wrap;  
    flex-direction: row;  
    align-content: stretch;  
    /* predeterminada */  
}
```

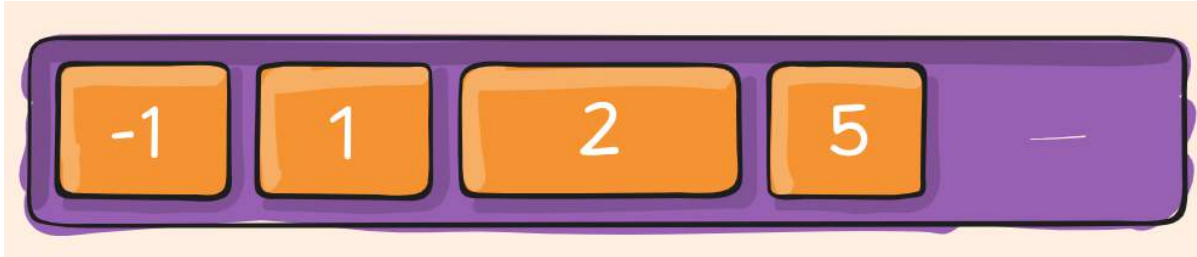
Propiedades para los flex items (hijos)



Order

Esta propiedad permite modificar el orden de aparición de un elemento. Recibe como valor números enteros.

```
.hijo-flex {  
  order: -1;  
}
```





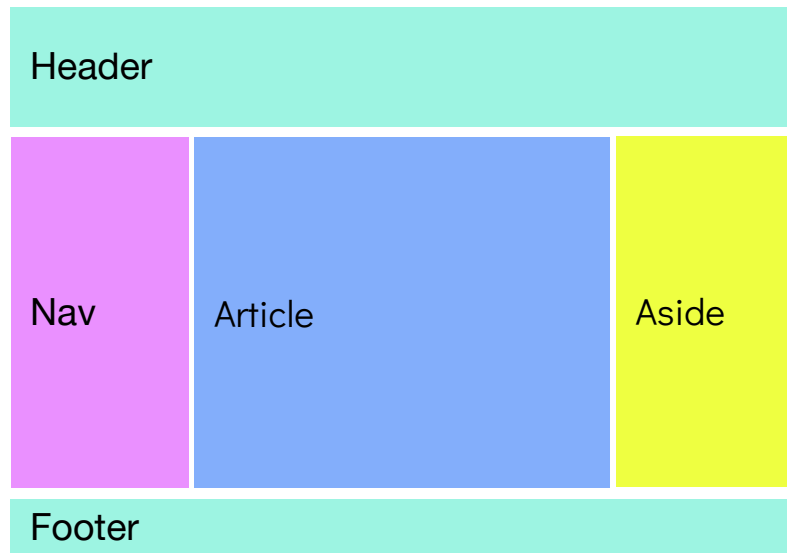
Ejemplo en vivo

¡Vamos a practicar lo visto!

Aplicamos Flexbox a un archivo HTML

PÁGINA CON FLEXBOX

Con lo aprendido en esta clase, podríamos hacer una estructura de página con la siguiente estructura, mediante flexbox:



Página con Flexbox

HTML

```
<header>Header</header>
<div id="main">
  <nav>Nav</nav>
  <aside>Aside</aside>
  <article>Article
</article>
</div>
<footer>Footer</footer>
```


Página con Flexbox

CSS

```
body{
  display: flex;
  flex-direction: column;
  height: 600px;
  padding: 1em;
}

main{
  display: flex;
  height: 300px;
}

article, nav, aside, header, footer {
  padding: 1rem;
  margin: 1px;
}
```

```
header, footer{
  background-color: #9df4e2;
}

header{
  height: 60px;
}

nav, aside{
  background-color: pink;
  width: 25%;
}

article{
  background-color: #83aefb;
  width: 50%;
}
```

Profundiza y conoce atajos en
nuestra [Cheat Sheet](#)



¿Preguntas?



¿Quieres saber más?
**Te dejamos material
ampliado de la clase**



MATERIAL AMPLIADO

Recursos multimedia

- ✓ [Imágenes de relleno](#) | placekitten
- ✓ [Imágenes de relleno](#) | placedog.net
- ✓ [Referencias de flexbox](#) | CSS Reference

Disponible en nuestro repositorio.

CODERHOUSE

Resumen de la clase hoy

- ✓ Modelo flexbox.
- ✓ Propiedades del contenedor padre.
- ✓ Propiedades de los ítems hijos.

Opina y valora
esta clase

Muchas gracias.

#DemocratizandoLaEducación