

Esta clase va a ser

- grabada

CODERHOUSE

Clase 05. DESARROLLO WEB

Flexbox

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.

CODERHOUSE

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.

CODERHOUSE

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

CODERHOUSE

Temario

04

CSS + Box Modeling

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

05

Flexbox

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

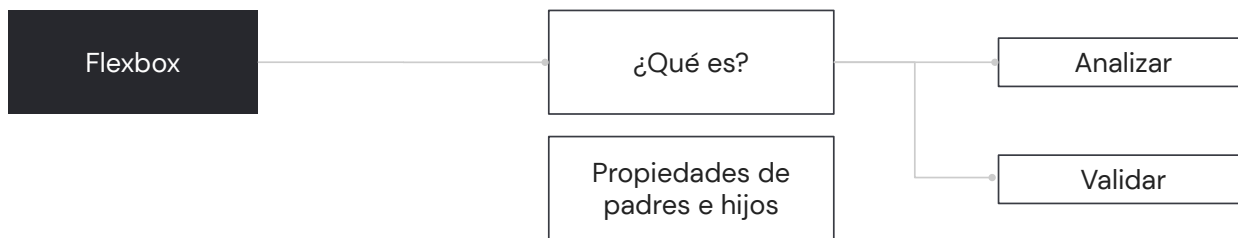
06

Grids

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

CODERHOUSE

MAPA DE CONCEPTOS



CODERHOUSE

Flexbox

CODERHOUSE

¿Qué es flexbox?

CODERHOUSE

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).

CODERHOUSE

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.

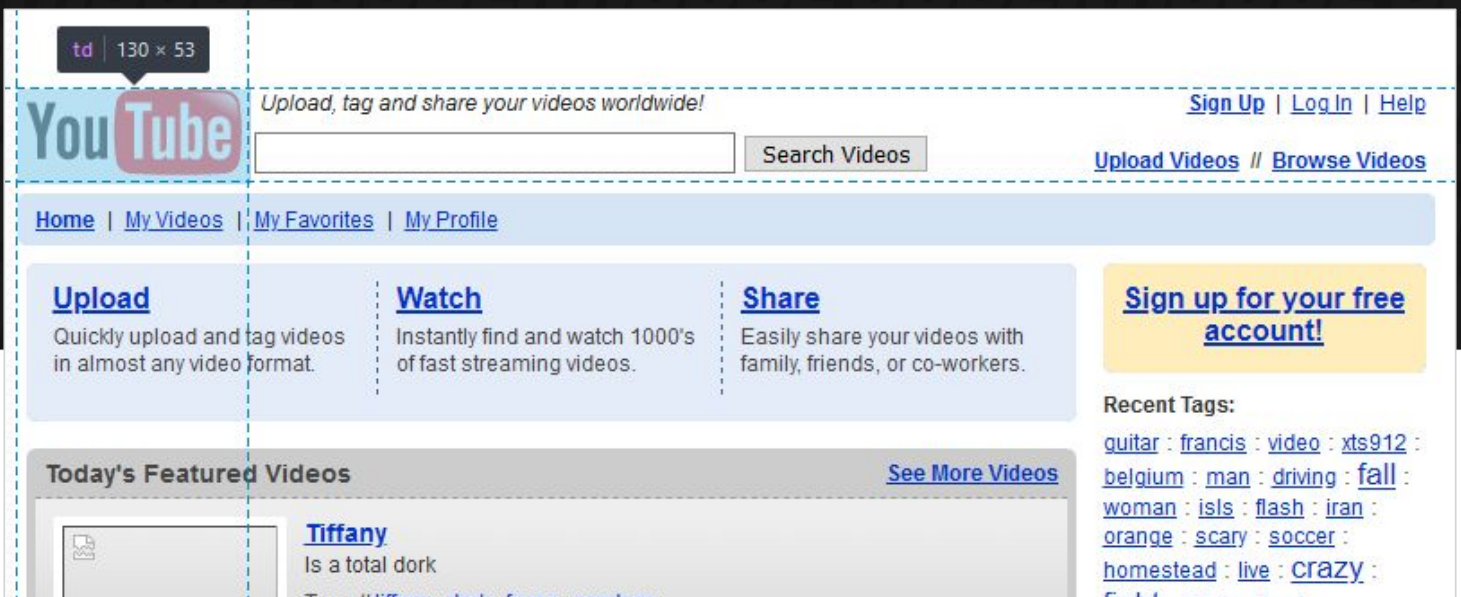
CODERHOUSE

Algo de historia

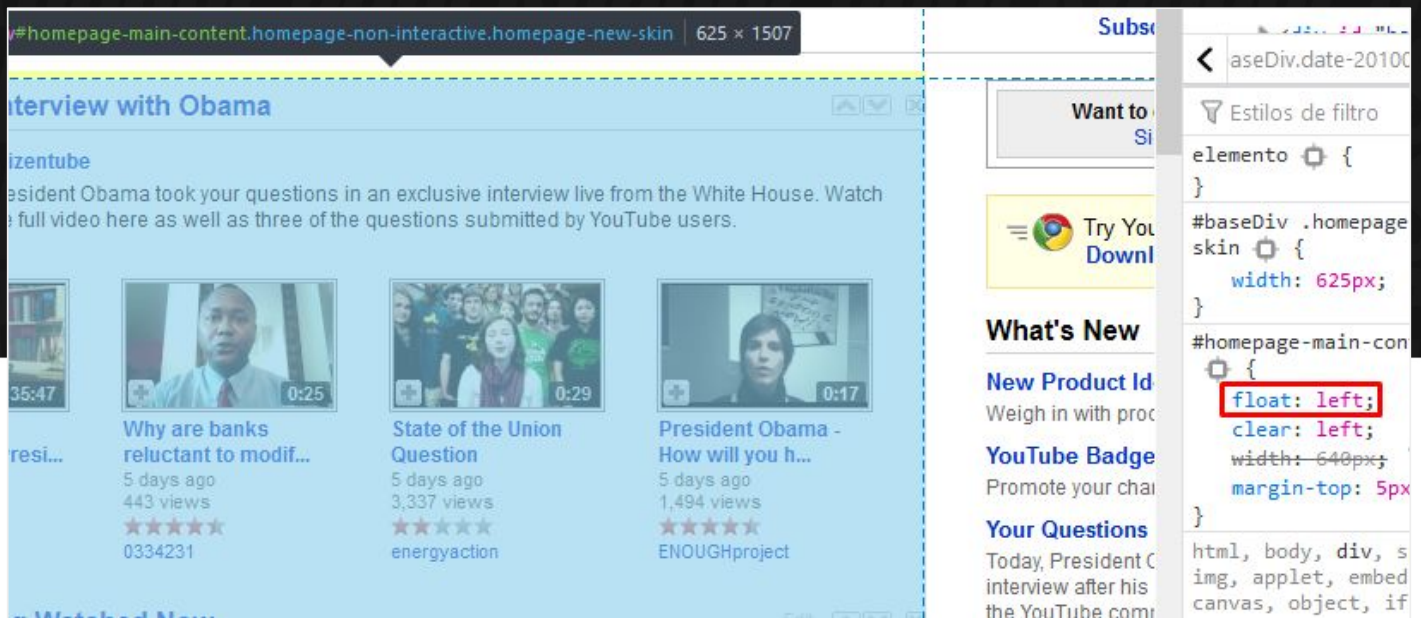
Veamos, brevemente, un poco de historia acerca del modo en que ha evolucionado la forma de crear estructuras...



Antes se usaban mucho las tablas



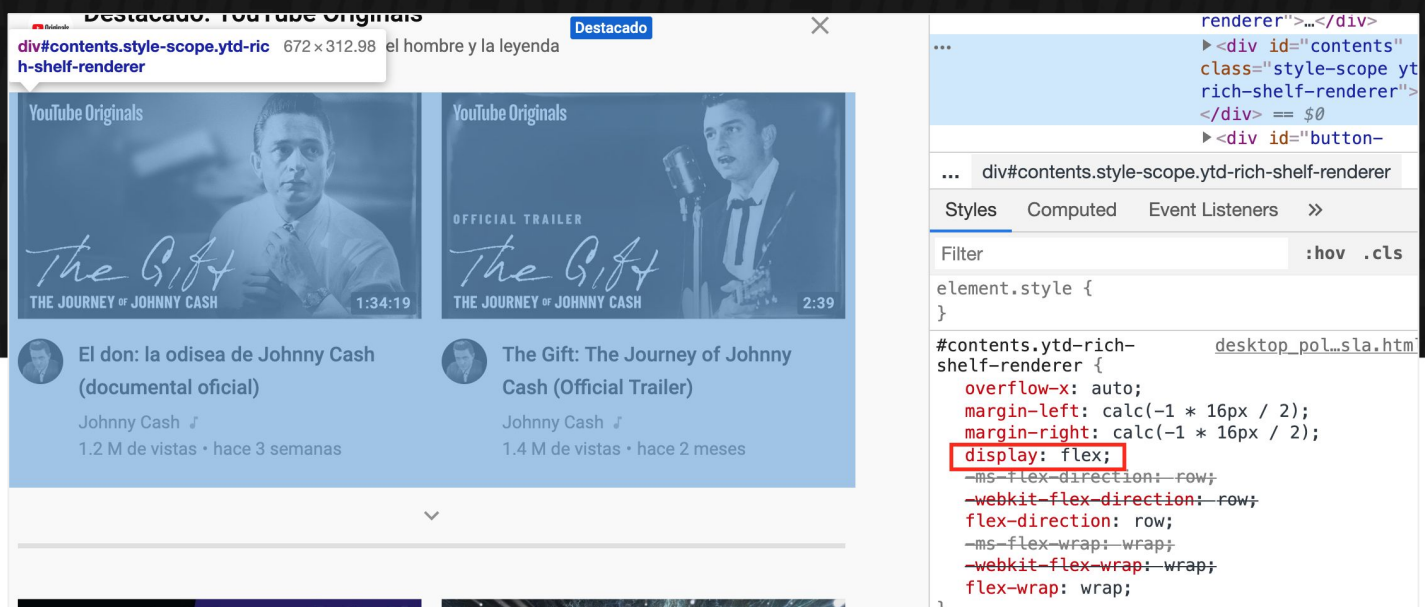
Algún tiempo atrás (no tanto)... se usaba el elemento Float



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

CODERHOUSE

En la actualidad se aplica Flexbox




Fuente: <http://www.youtube.com/>

CODERHOUSE



Ejemplo en vivo

¡Vamos a practicar lo visto!

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

CODERHOUSE



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.**

CODERHOUSE

Propiedades de padres e hijos

CODERHOUSE

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.

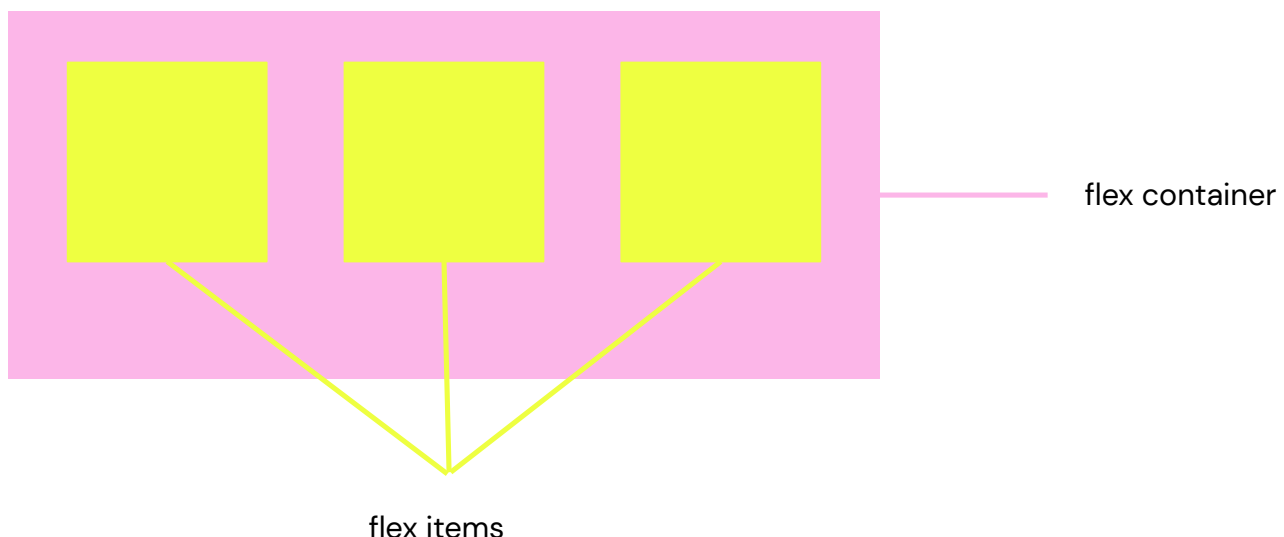
CODERHOUSE

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.

CODERHOUSE

Propiedades del padre



CODERHOUSE

¿Cómo empezamos?

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

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

CODERHOUSE

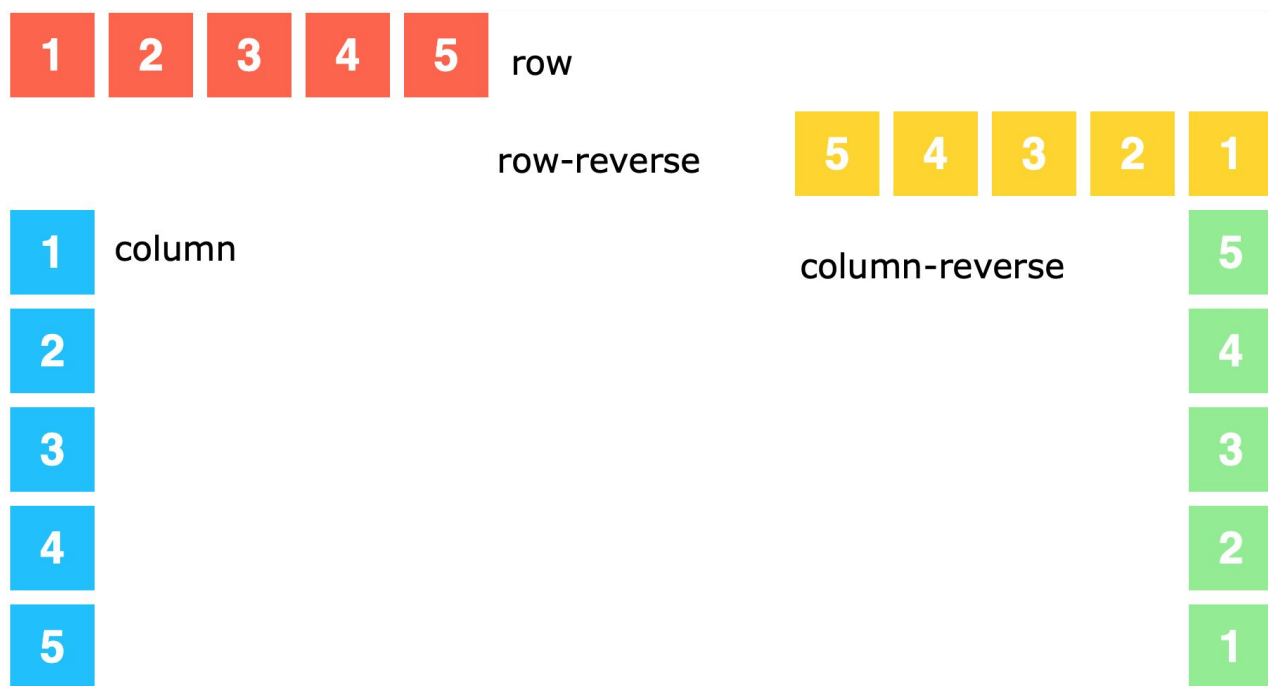
Flex-direction

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 */  
}
```

CODERHOUSE



Resumen de Flex-direction

Flex-wrap

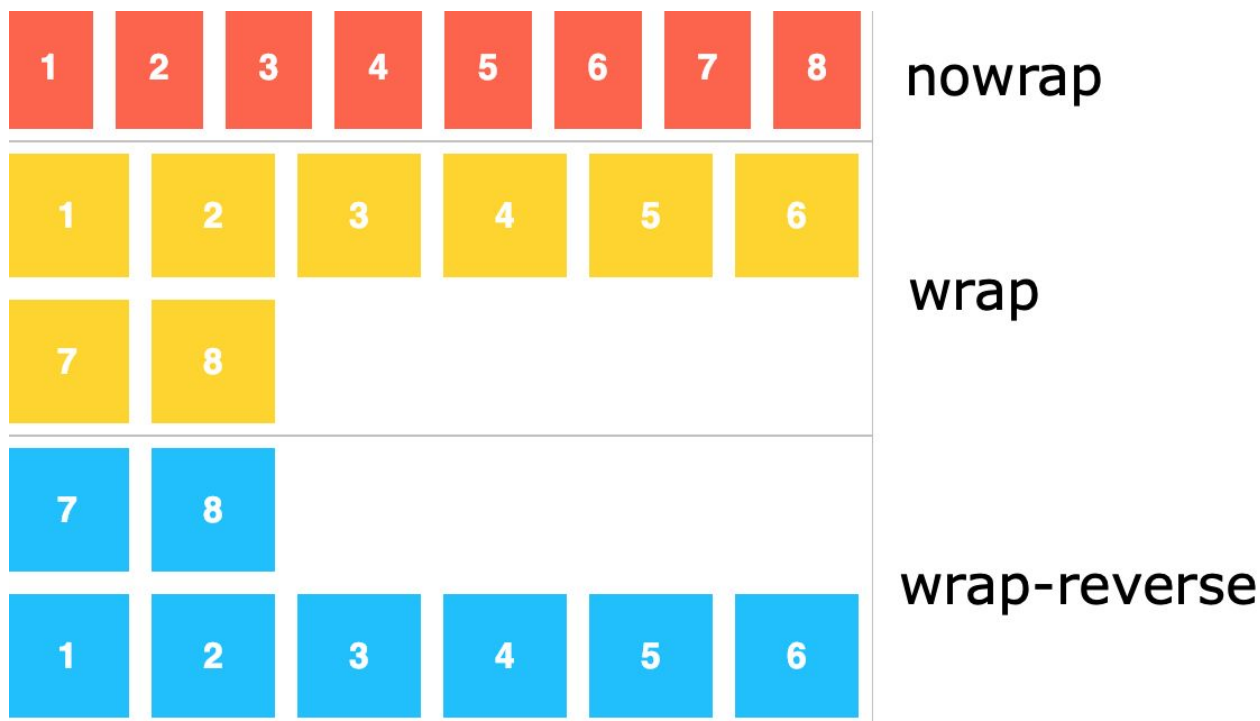
El comportamiento inicial del contenedor flexible es poder mantener los flex items en su eje, sin importar que las dimensiones de los mismos cambien.

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 */  
}
```



Resumen de Flex-wrap

CODERHOUSE

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;  
}
```

CODERHOUSE



Break

¡10 minutos y volvemos!

CODERHOUSE

¿Preguntas?

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

CODERHOUSE

Justify-content

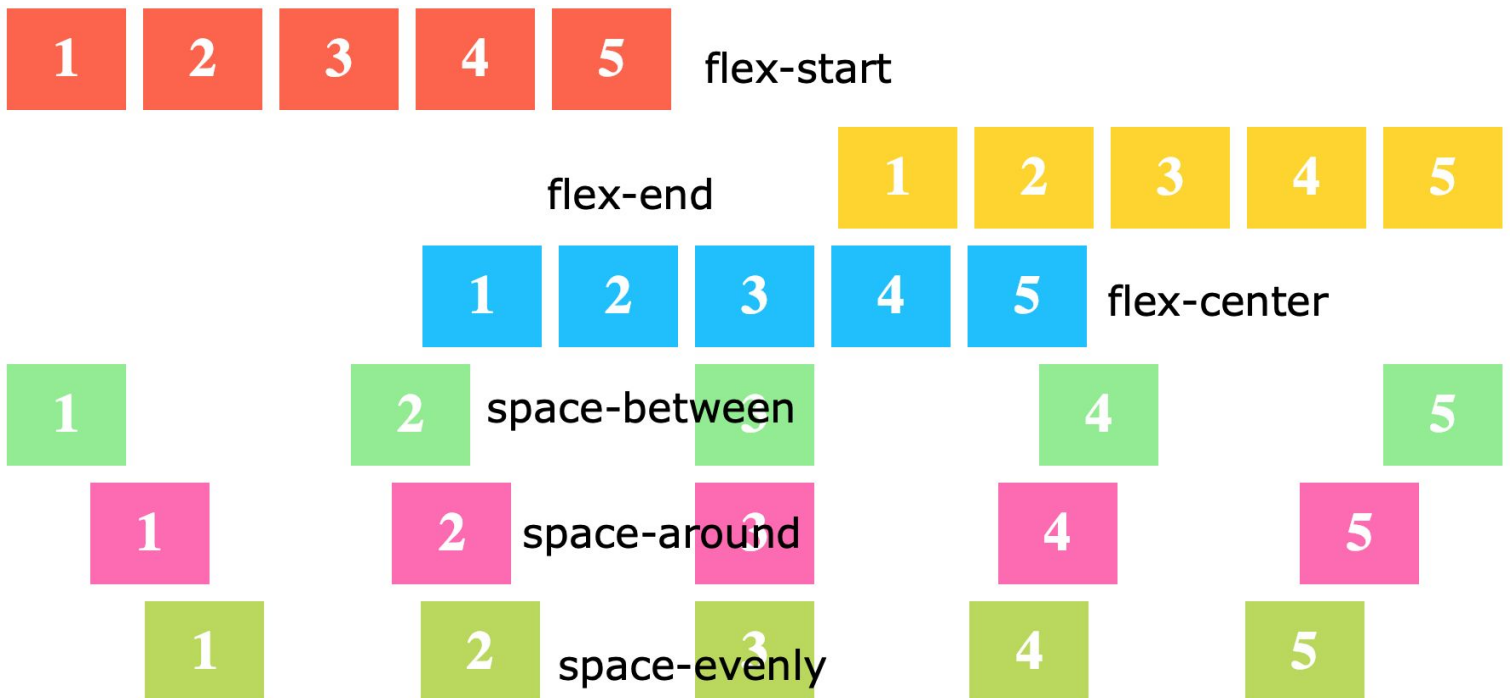
Justify-content nos va a permitir **alinear los elementos en el eje principal**. Este puede ser el eje vertical u horizontal, según lo especifiquemos con flex-direction. Con esta propiedad podemos distribuir los flex items (hijos) en el contenedor (padre), cuando los ítems no utilicen todo el espacio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

CODERHOUSE



Justify-content resumen

CODERHOUSE

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

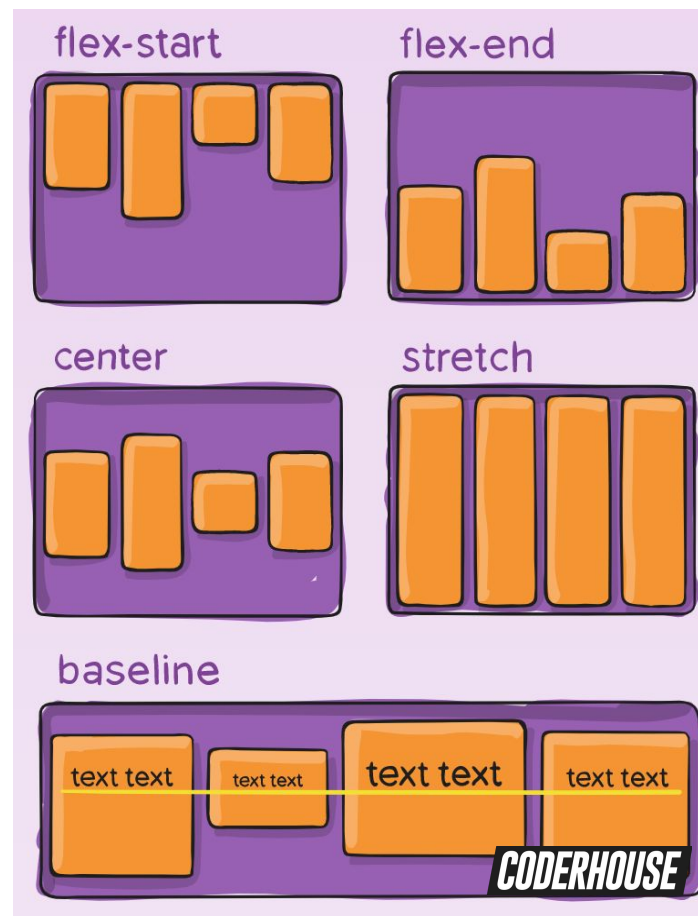
Sus valores posibles son:

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

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

CODERHOUSE

Align-items resumen



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.

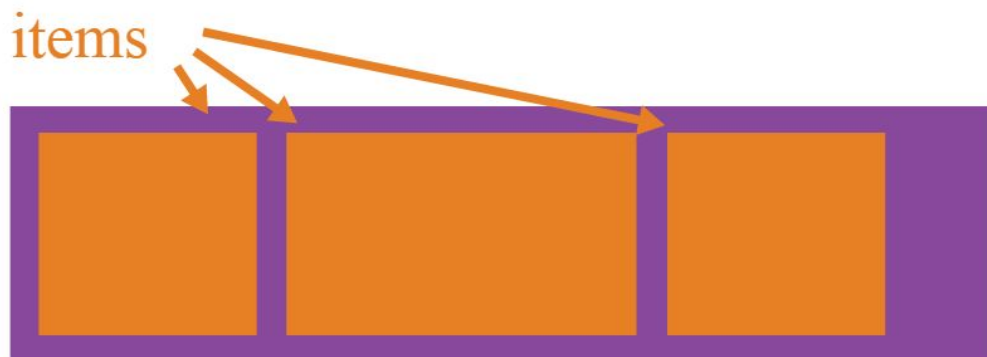
Su valores posibles son:

- `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 */  
}
```

CODERHOUSE

Propiedades para los flex items (hijos)



CODERHOUSE

Order

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

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



CODERHOUSE



Ejemplo en vivo

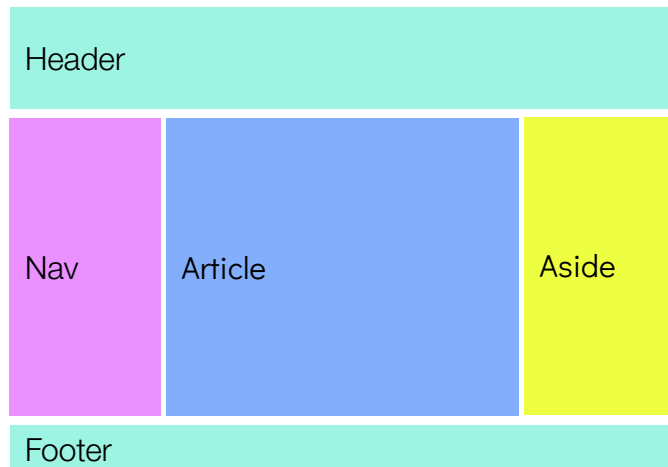
¡Vamos a practicar lo visto!

Aplicamos Flexbox a un archivo HTML

CODERHOUSE

PÁGINA CON FLEXBOX

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



CODERHOUSE

Página con Flexbox

HTML

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

CODERHOUSE

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%;
}
```

CODERHOUSE

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



CODERHOUSE

¿Preguntas?

CODERHOUSE



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

CODERHOUSE

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.

CODERHOUSE

Opina y valora
esta clase

CODERHOUSE

Muchas gracias.

CODERHOUSE

#DemocratizandoLaEducación

CODERHOUSE