



**¡Les damos la  
bienvenida!**

¿Comenzamos?

Esta clase va a ser

- grabada

Clase 05. DESARROLLO WEB

# Flexbox

# Temario

04

## CSS+ Box Modeling

- ✓ Bem
- ✓ Box Modeling
- ✓ Display
- ✓ Posiciones

05

## Flexbox

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

02

## Grids

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

# Objetivos de la clase

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

# Glosario

**Box model:** el concepto de que “todo es una caja”, da lugar a algo llamado en web como 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.

## **Espacio exterior**

**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.

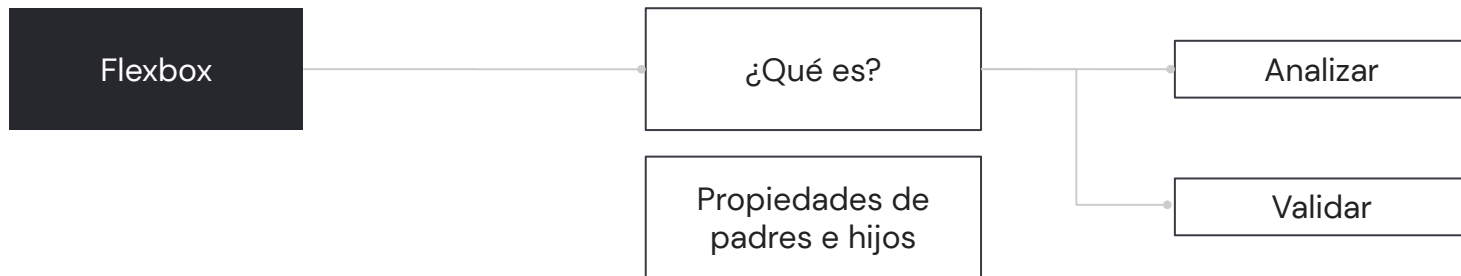
CLASE N°04

# 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.

## 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. Como vimos en la clase anterior, utilizando la propiedad float podíamos desplazar contenedores.

Con Flexbox ya no necesitarás usar float. Podrás posicionar y distribuir los elementos como tú quieras.

# ¿Qué es flexbox?

- ✓ Es un conjunto de propiedades de CSS.
- ✓ Se basa sobre un contenedor (padre) para ordenar a sus ítems (hijos).
- ✓ No solo se podrá posicionar elementos vertical y horizontalmente, sino que podrás 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)

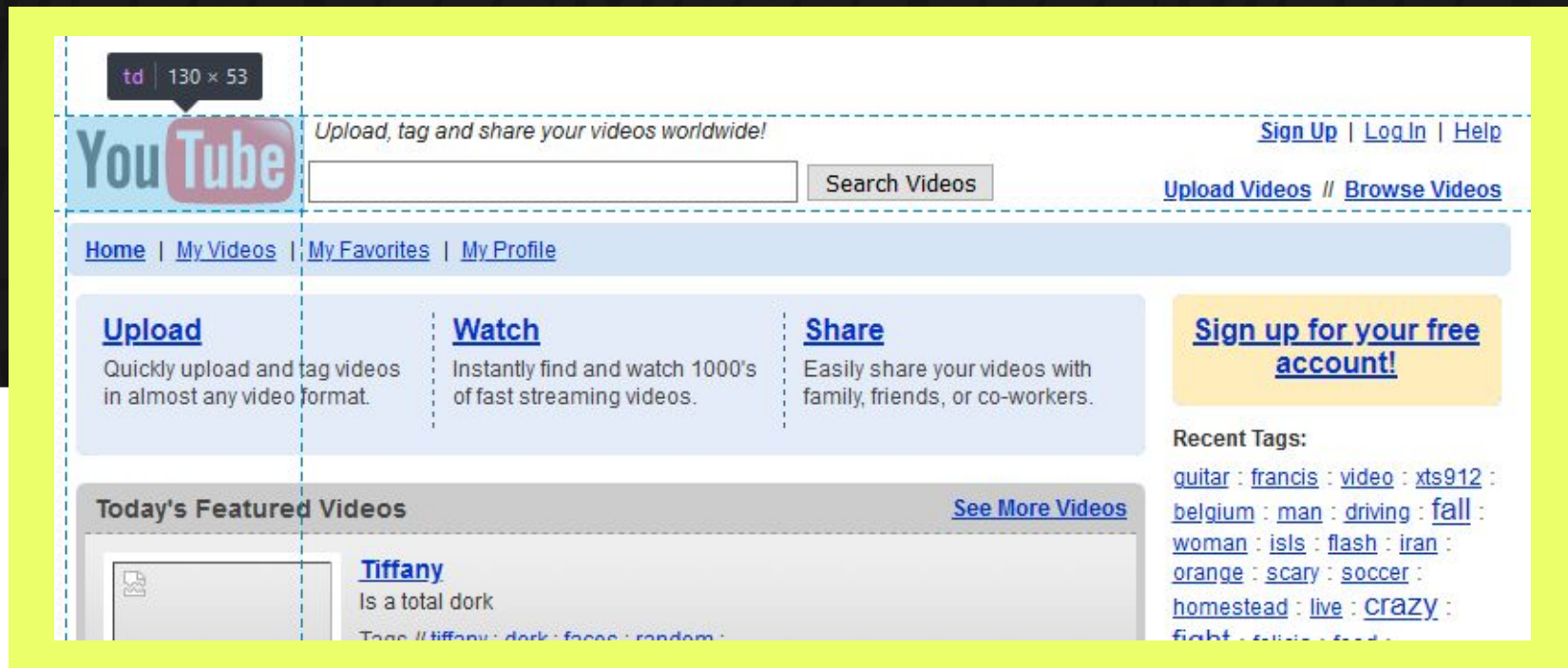
# Beneficios

- ✓ 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

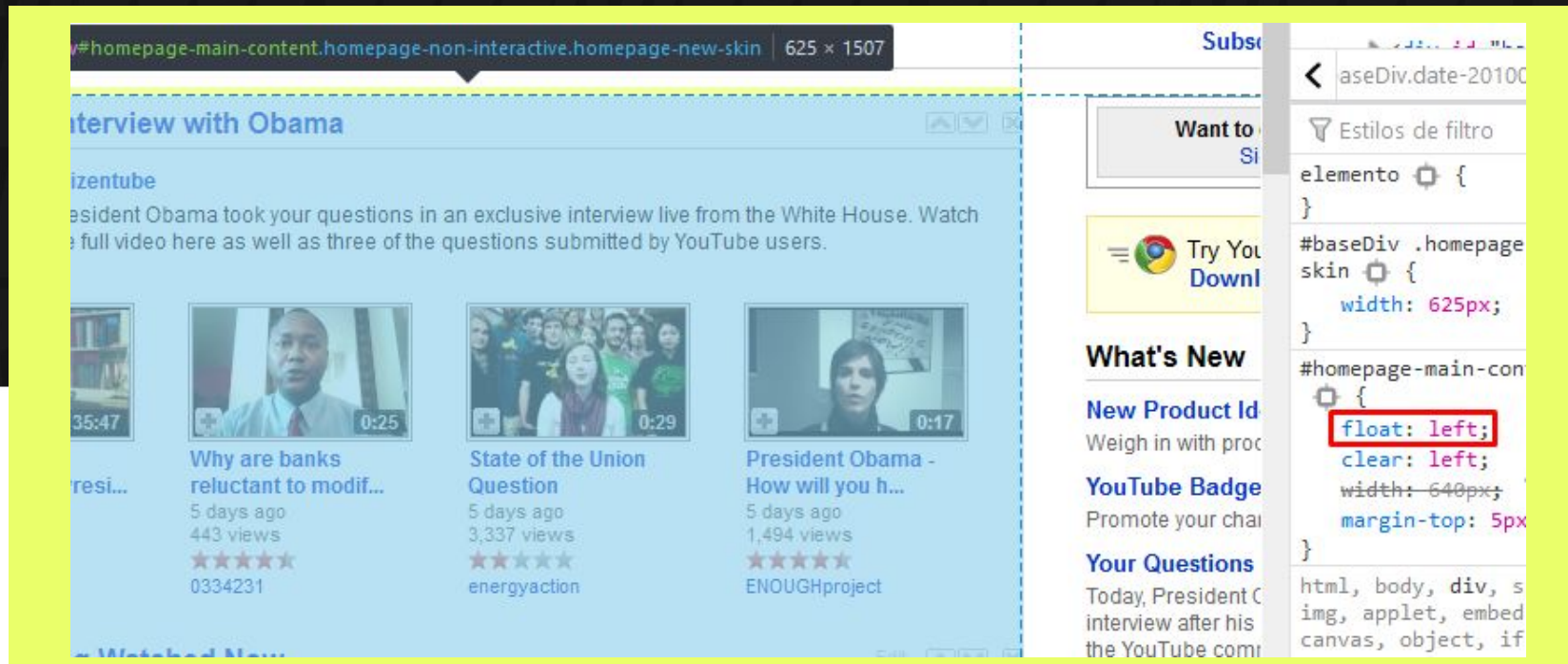
Veamos, brevemente, un poco de historia acerca del modo en que ha evolucionado la forma de crear estructuras...

# Antes se usaban mucho las tablas



Fuente: <https://web.archive.org/web/20050810030357/http://www.youtube.com/>

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





# En la actualidad se aplica Flexbox

Destacado: YouTube Originals

div#contents.style-scope.ytd-rich-shelf-renderer 672 x 312.98 el hombre y la leyenda

Destacado

YouTube Originals

*The Gift*  
THE JOURNEY OF JOHNNY CASH 1:34:19

El don: la odisea de Johnny Cash (documental oficial)  
Johnny Cash 1.2 M de vistas • hace 3 semanas

YouTube Originals

OFFICIAL TRAILER  
*The Gift*  
THE JOURNEY OF JOHNNY CASH 2:39

The Gift: The Journey of Johnny Cash (Official Trailer)  
Johnny Cash 1.4 M de vistas • hace 2 meses

```
renderer">...</div>
...
  <div id="contents"
    class="style-scope yt
      rich-shelf-renderer">
    </div> == $0
  <div id="button-
... div#contents.style-scope.ytd-rich-shelf-renderer

Styles Computed Event Listeners >>

Filter :hov .cls

element.style {
}

#contents.ytd-rich-shelf-renderer {
  overflow-x: auto;
  margin-left: calc(-1 * 16px / 2);
  margin-right: calc(-1 * 16px / 2);
  display: flex;
  -ms-flex-direction: row;
  -webkit-flex-direction: row;
  flex-direction: row;
  -ms-flex-wrap: wrap;
  -webkit-flex-wrap: wrap;
  flex-wrap: wrap;
}
```

# Ejemplos

# Páginas con flexbox

The image displays a screenshot of a YouTube homepage on the left and its corresponding browser developer tools on the right, illustrating the use of Flexbox in web design.

**YouTube Homepage:**

- Header:** Includes the YouTube logo, a search bar, and a navigation menu with icons for home, trends, subscriptions, library, and history.
- Channel Card:** A card for "Pasión Por La Roja" with 676,012 views and a video thumbnail.
- Section Header:** "Conciertos desde casa" (Home Concerts) with a "Destacado" (Featured) badge.
- Video Grid:** A grid of video thumbnails, including "Priceless Experiences at Home" by Camila Cabello and "Coke Studio Sessions: Katy Perry".

**Browser Developer Tools:**

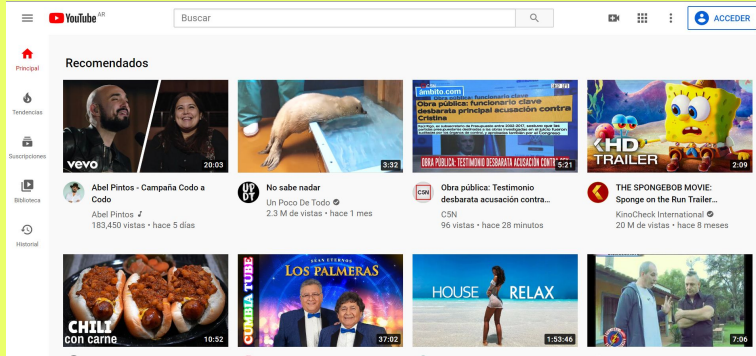
- Elements Panel:** Shows the DOM tree with the following structure:

```
<ytd-browse>  
  <ytd-channel-legal-info-renderer>  
  <ytd-playlist-sidebar-renderer class="style-scope ytd-browse disable-upgrade hidden">  
  <ytd-playlist-sidebar-renderer>  
  <ytd-settings-sidebar-renderer class="style-scope ytd-browse disable-upgrade hidden">  
  <ytd-settings-sidebar-renderer>  
  <ytd-two-column-browse-results-renderer class="style-scope ytd-browse grid grid-disabled" page-subtype="home"> == $0  
    <!--css-build:shady-->  
    <div id="primary" class="style-scope ytd-two-column-browse-results-renderer">
```
- Styles Panel:** Shows the CSS rules for the selected element, including:

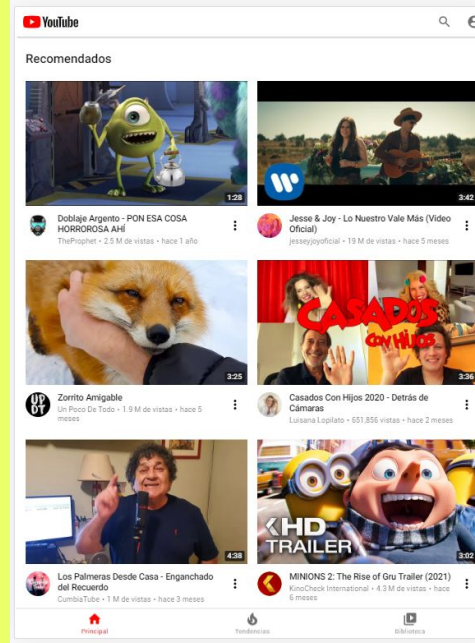
```
margin-bottom: 16px;  
width: 100%;  
display: flex;  
flex-direction: column-reverse;  
justify-content: flex-end;
```
- Computed Tab:** Shows the computed styles for the element, including:

```
width: 746.667px  
height: 4912.530px  
padding: 16px
```

# Páginas con flexbox



Desktop



Tablet



Mobile

# Páginas con flexbox

The image shows a screenshot of the Amazon Argentina homepage, illustrating the use of Flexbox in web design. The page is divided into several sections:

- Header:** Includes the Amazon logo, a search bar, and navigation links like "Hola, Identifícate Cuenta y Listas" and "Devoluciones y Pedidos".
- Sub-header:** Features a banner with the text "Estamos entregando a tu región con opciones de envío limitadas. Por favor anticipa un periodo de entrega prolongado. Más información."
- Main Content:** Consists of three columns of product categories, each with a grid of images and a "Ver más" link:
  - Compra por Categoría:** A 2x2 grid showing "Computadoras", "Videojuegos", "Bebé", and "Juguetes y Juegos".
  - AmazonBasics:** A single image of a charging cable with a "Ver más" link.
  - Electrónicos:** A single image of a camera and headphones with a "Ver más" link.

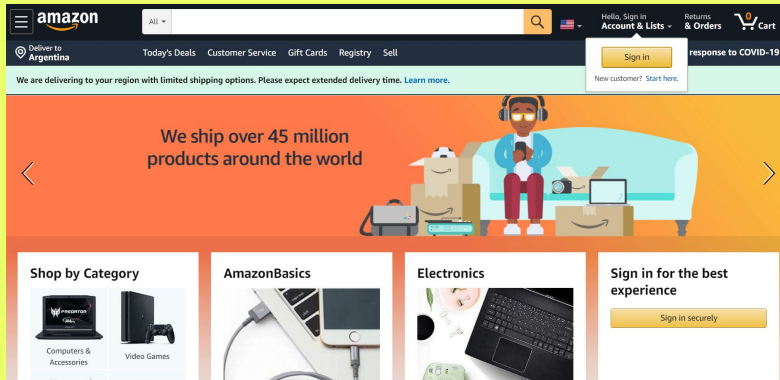
On the right side, the browser's developer tools are open, showing the CSS styles for the product cards. The styles include:

```
margin-bottom: 0;
margin-top: 0;
overflow: hidden;
padding-top: 20px;

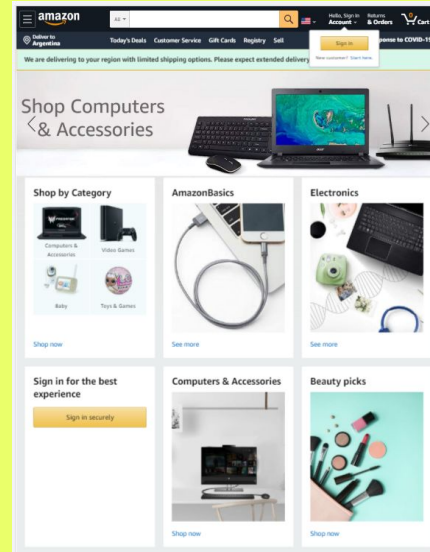
.fluid-card {
  display: flex;
  -webkit-box-orient: vertical;
  -webkit-box-direction: normal;
  flex-direction: column;
}

.a-cardui {
  margin-top: 20px;
  margin-bottom: 20px;
  background-color: #fff;
  overflow: auto;
  padding: 20px 0 15px 0;
```

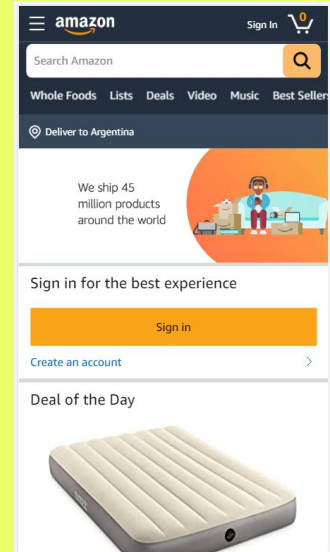
# Páginas con flexbox



Desktop



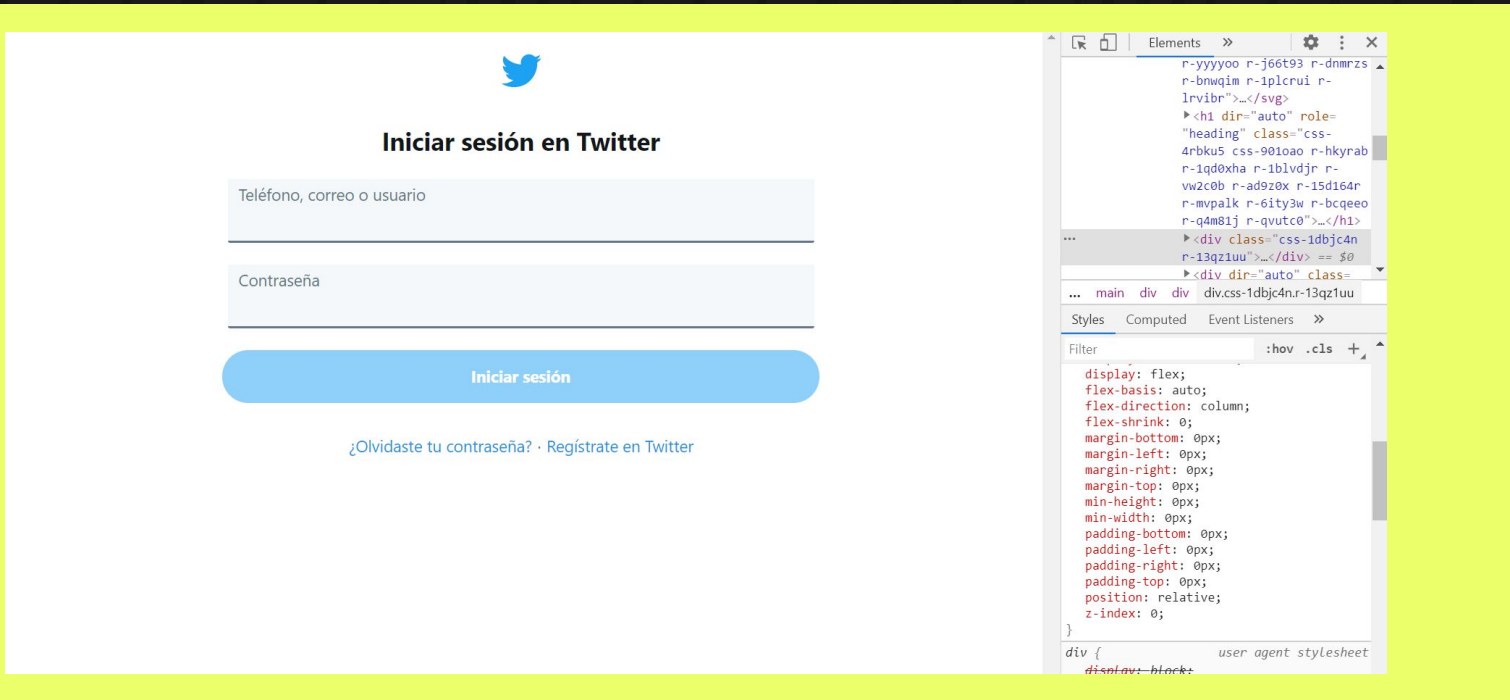
Tablet



Mobile



# Páginas con flexbox



The image shows a Twitter login page and its browser developer tools. The page has a white background with a blue Twitter bird logo at the top center. Below the logo is the text "Iniciar sesión en Twitter". There are two input fields: "Teléfono, correo o usuario" and "Contraseña". Below these is a blue button labeled "Iniciar sesión". At the bottom, there is a link: "¿Olvidaste tu contraseña? · Regístrate en Twitter". The browser developer tools are open on the right, showing the "Elements" panel with the HTML structure and the "Styles" panel with the CSS rules for the selected element.

**Iniciar sesión en Twitter**

Teléfono, correo o usuario

Contraseña

Iniciar sesión

¿Olvidaste tu contraseña? · Regístrate en Twitter

**Elements**

```
r-yyyyoo r-j66t93 r-dnmrzs
r-bnwqim r-1plcrui r-
lrvihr">...</svg>
<h1 dir="auto" role=
"heading" class="css-
4rbku5 css-901oao r-hkyrab
r-1qd0xha r-1blvdjr r-
vw2cob r-ad920x r-15d164r
r-mvpalk r-6ity3w r-bcqeeo
r-q4m81j r-qvutc0">...</h1>
...
<div class="css-1dbjc4n
r-13qz1uu">...</div> == $0
<div dir="auto" class=
... main div div
div.css-1dbjc4n.r-13qz1uu
```

**Styles** Computed Event Listeners

Filter :hov .cls +

```
display: flex;
flex-basis: auto;
flex-direction: column;
flex-shrink: 0;
margin-bottom: 0px;
margin-left: 0px;
margin-right: 0px;
margin-top: 0px;
min-height: 0px;
min-width: 0px;
padding-bottom: 0px;
padding-left: 0px;
padding-right: 0px;
padding-top: 0px;
position: relative;
z-index: 0;
}
div {
  display: block;
}
```

# Propiedades de padres e hijos



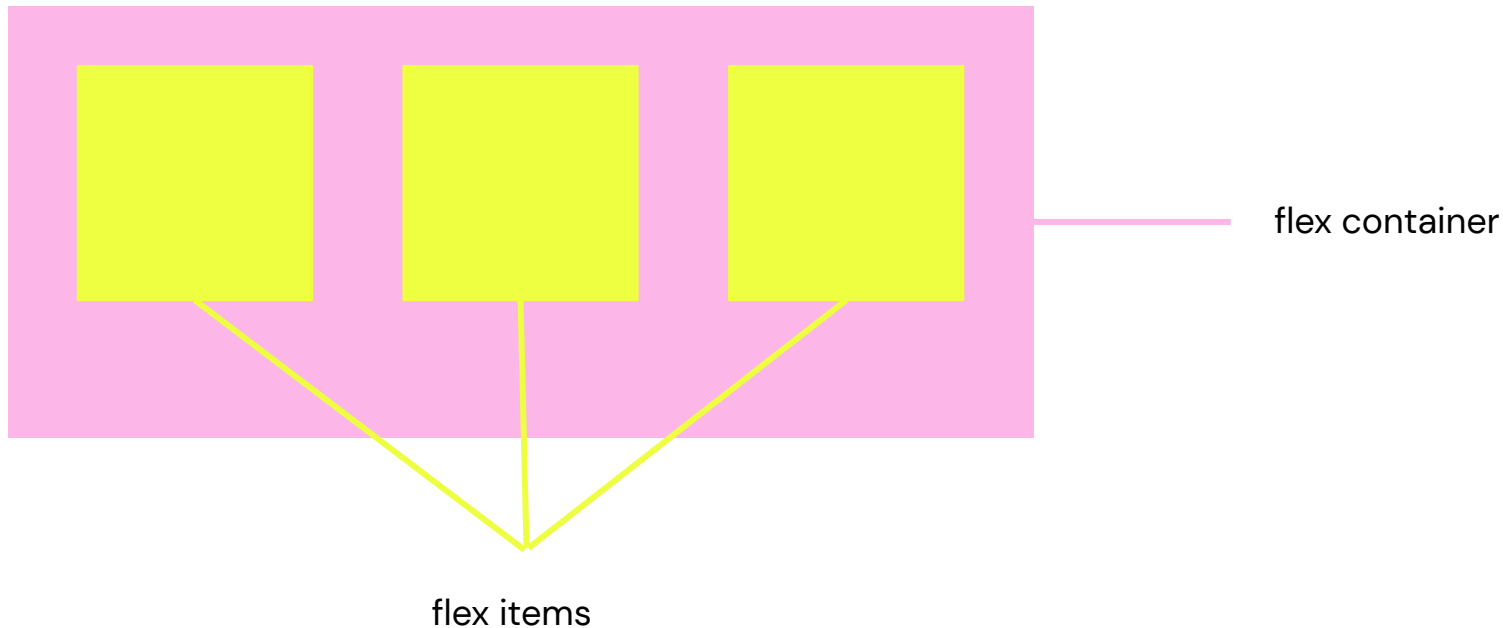
# Propiedades para aplicar en el contenedor flexible (el padre)

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

# Propiedades para aplicar en el contenedor flexible (el padre)

<b>justify-content</b>	Alinear horizontalmente a los hijos si el padre es "fila" o verticalmente si el padre es "columna".
<b>align-items</b>	Alinear verticalmente a los hijos (si el padre es "columna").
<b>align-content</b>	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: row

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

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



# Flex-direction: row-reverse

Con el valor row-reverse (fila inversa) los flex items se apilan en una fila de derecha a izquierda.

```
.padre-flex {  
  display: flex;  
  flex-direction: row-reverse;  
}
```



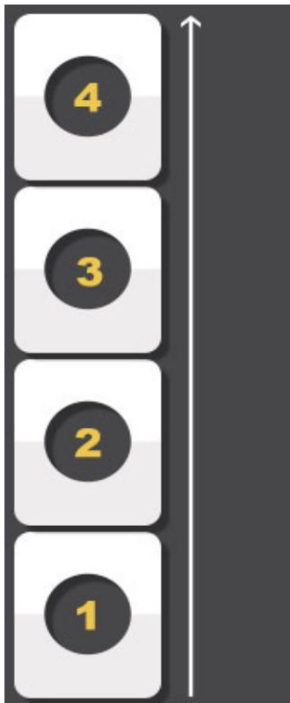
# Flex-direction: column



Con el valor `column`, los flex items se apilan en una columna de arriba hacia abajo.

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

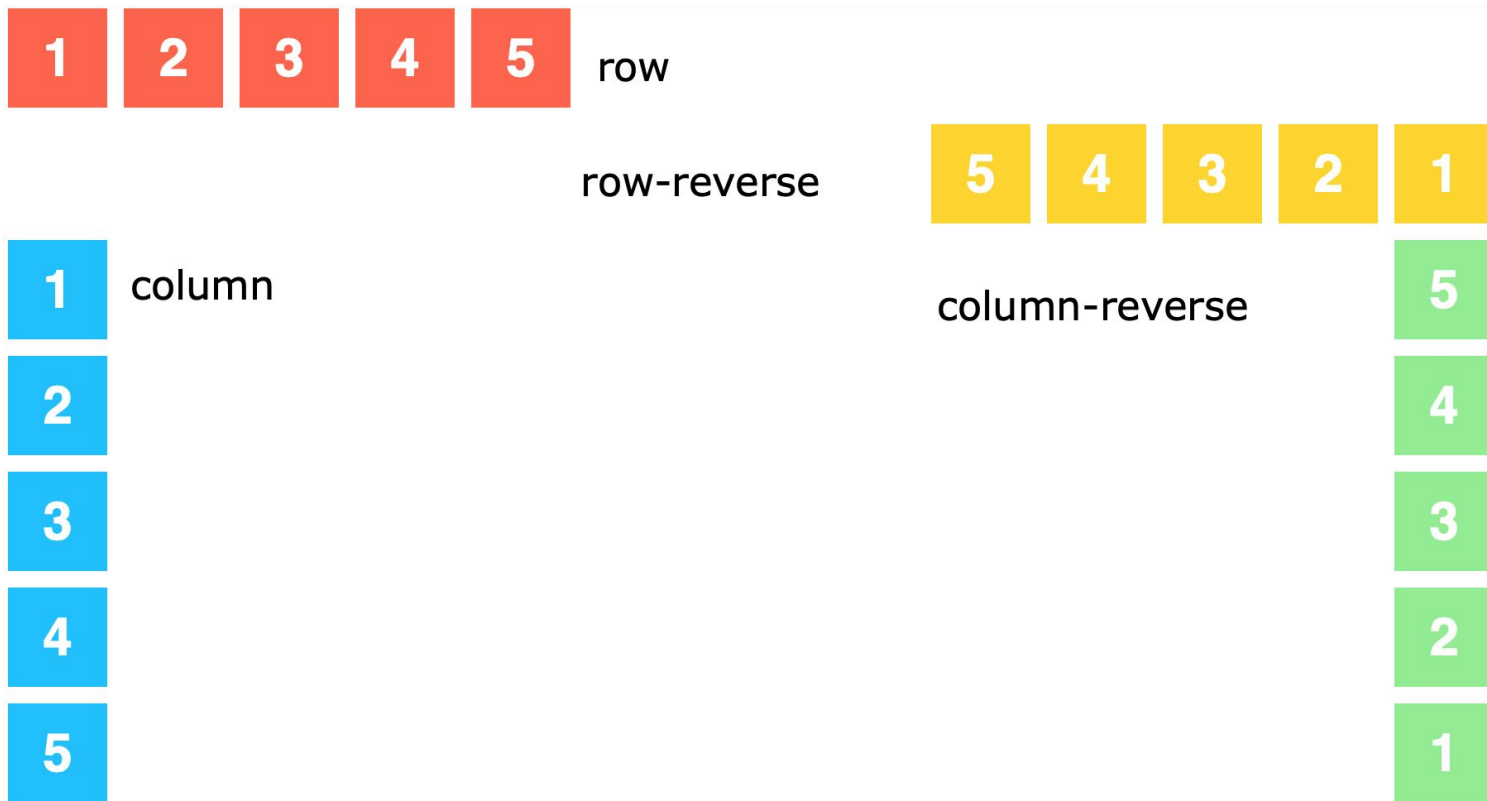
# Flex-direction: column



Con el valor `column-reverse` los flex items se apilan en una columna de abajo hacia arriba.

```
.padre-flex {  
  display: flex;  
  flex-direction: column-reverse;  
}
```





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.**



# Flex-wrap: nowrap

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



# Flex-wrap: wrap

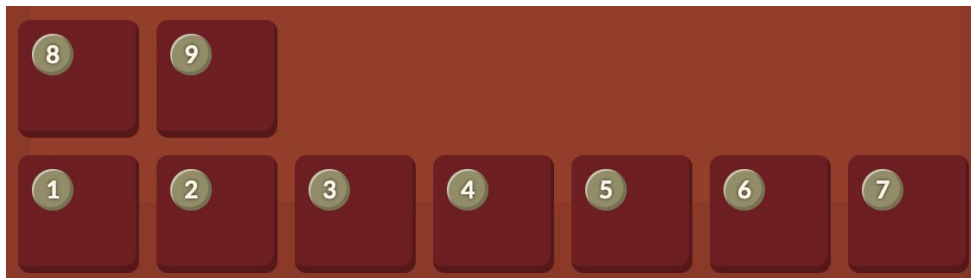
Los flex items (hijos) pueden romper la línea del eje horizontal, si les es necesario para conservar las características de sus dimensiones. Esto es de izquierda a derecha, y de arriba a abajo.



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

# Flex-wrap: wrap-reverse

Esta vez el orden es de izquierda a derecha, y de abajo a arriba.



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



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 la propiedad de flex-direction, y luego la de flex-wrap.

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

# Flex-wrap

- ✓ Justify-content nos va a permitir alinear los elementos.
- ✓ Esto puede ser de forma vertical u horizontal, según lo especifiquemos con flex-direction.
- ✓ Nos va a ayudar a distribuir los flex items (hijos) en el contenedor (padre), cuando los ítems no utilicen todo el espacio disponible en su eje principal actual.





# Justify-content

Los siguientes ejemplos parten de la base del contenedor en "row":

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



# Justify-content: flex-start

Consiste en alinear los flex items (hijos) al lado izquierdo.

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





# Justify-content: flex-end

Consiste en alinear los flex items (hijos) al lado derecho.

```
.padre-flex {  
  display: flex; flex-direction: row;  
  justify-content: flex-end;  
}
```





# Justify-content: center

Consiste en alinear los flex items (hijos) al centro.

```
.padre-flex {  
  display: flex; flex-direction: row;  
  justify-content: center;  
}
```





# Justify-content: space-between

Es hacer que los flex items (hijos) tomen la misma distancia o espaciado entre ellos dentro del contenedor flexible, quedando el primer y último elemento pegados con los bordes del contenedor en el eje principal.

```
.padre-flex {  
  display: flex; flex-direction: row;  
  justify-content: space-between;  
}
```





# Justify-content: space-around

Muestra los flex items (hijos) con el mismo espacio de separación entre sí. El espaciado entre los bordes lo toman del contenedor padre.

```
.padre-flex {  
  display: flex;  flex-direction: row;  
  justify-content: space-around;  
}
```





# Justify-content: space-evenly

Hace que el espacio entre los flex items (hijos) sea igual. No es lo mismo que space-around.

```
.padre-flex {  
  display: flex; flex-direction: row;  
  justify-content: space-evenly;  
}
```





Justify-content resumen





# Justify-content

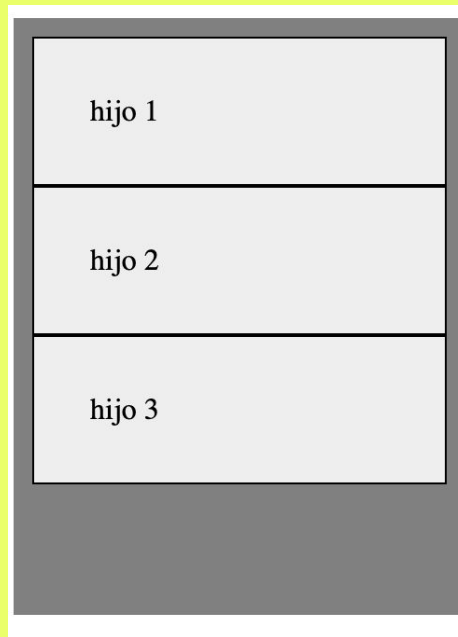
Los siguientes ejemplos parten de la base del contenedor en "column", con altura definida:

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



# Justify-content: flex-start

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
flex-start;  
  height: 300px;  
}
```





# Justify-content: flex-end

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-end;  
  height: 300px;  
}
```





# Justify-content: center

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  height: 300px;  
}
```

hijo 1

hijo 2

hijo 3



# Justify-content: space-between

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
space-between;  
  height: 300px;  
}
```





# Justify-content: space-around

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
space-around;  
  height: 300px;  
}
```

hijo 1

hijo 2

hijo 3



# Justify-content: space-around

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
space-around;  
  height: 300px;  
}
```





# Justify-content: space-around

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
space-around;  
  height: 300px;  
}
```







# Justify-content: space-around

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
space-around;  
  height: 300px;  
}
```





# Justify-content: space-evenly

```
.padre-flex {  
  display: flex;  
  flex-direction: column;  
  justify-content:  
space-evenly;  
  height: 300px;  
}
```

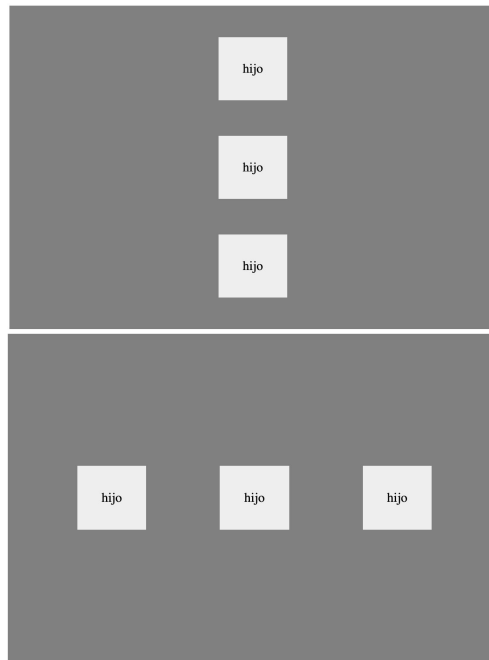
hijo 1

hijo 2

hijo 3

# Align-items

- ✓ Alinear los elementos verticales de forma horizontal.  
(flex-direction: column)
- ✓ Alinear los elementos horizontales de forma vertical.  
(flex-direction: row)

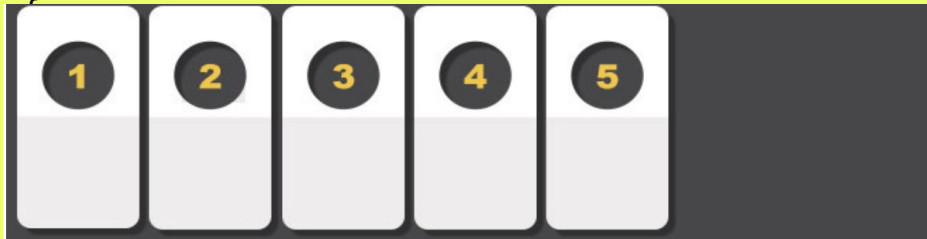




# Align-items: stretch

Tratará de llenar toda la altura (o anchura) del contenedor, siempre y cuando los hijos no tengan propiedades de dimensión definidas.

```
.padre-flex {  
  display: flex; flex-direction: row;  
  align-items: stretch;  
}
```





# Align-items: flex-start

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





# Align-items: flex-end

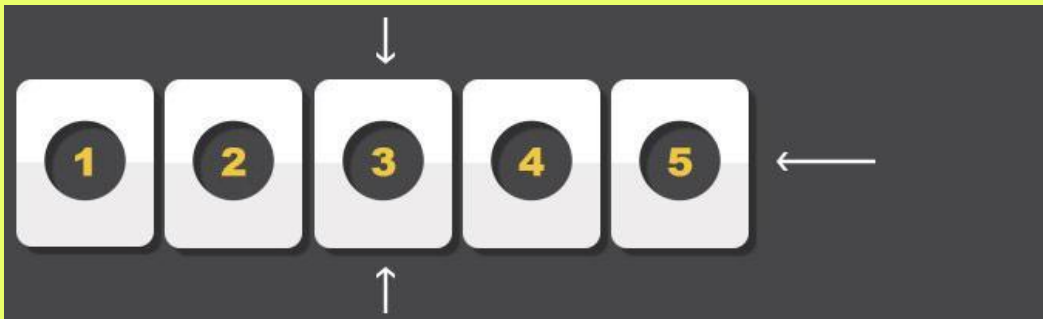
```
.padre-flex {  
  display: flex; flex-direction: row;  
  align-items: flex-end; /* predeterminado */  
}
```





# Align-items: center

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





# Align-items: baseline

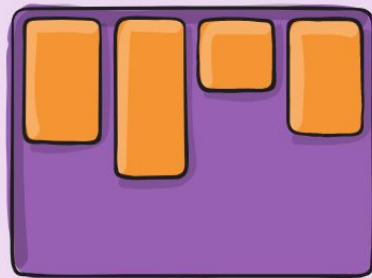
```
.padre-flex {  
  display: flex; flex-direction: row;  
  align-items: baseline; /* predeterminado */  
}
```



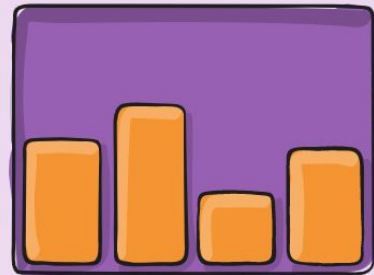


# Align-items resumen

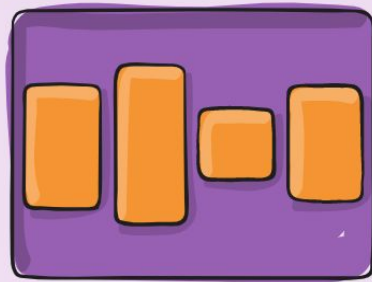
flex-start



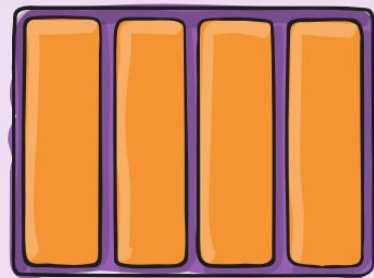
flex-end



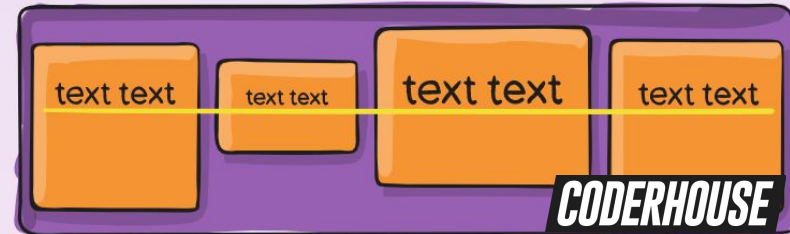
center



stretch



baseline

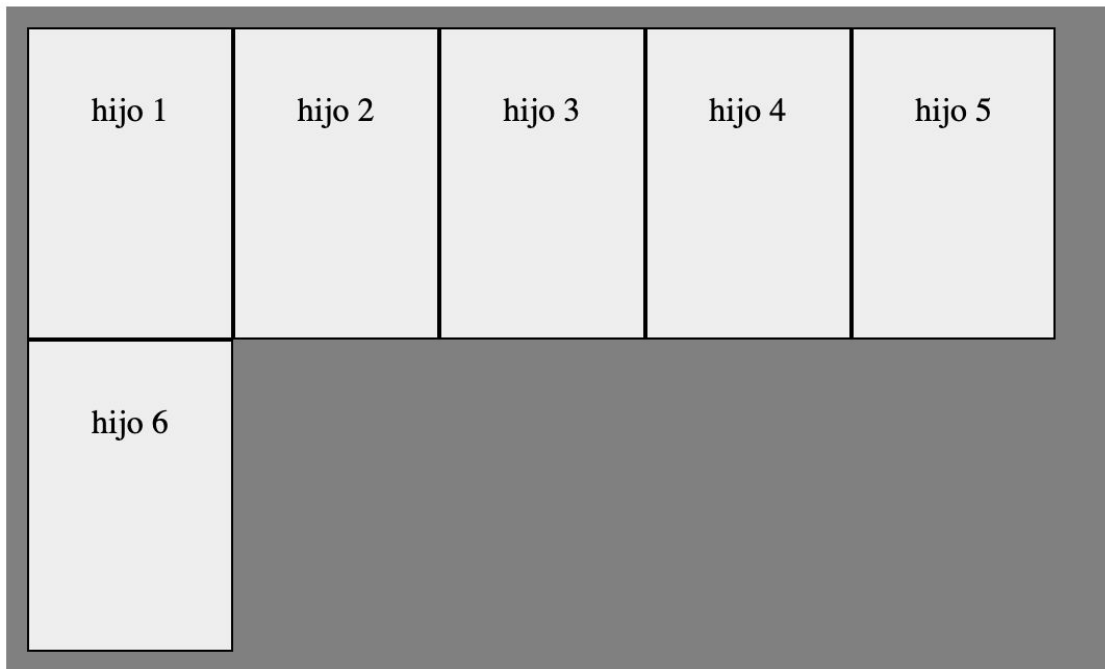


# Align-content

- ✓ 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.

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

# Align-content: stretch



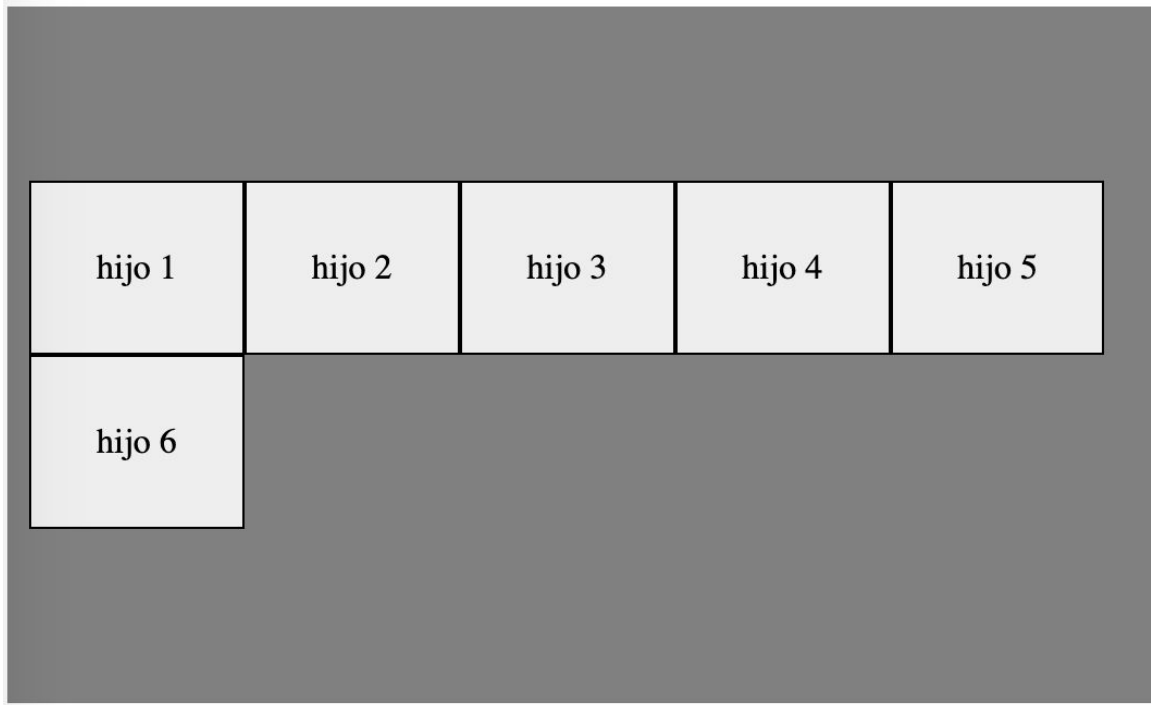
# Align-content: flex-start



# Align-content: flex-end



# Align-content: center



# Align-content: space-between

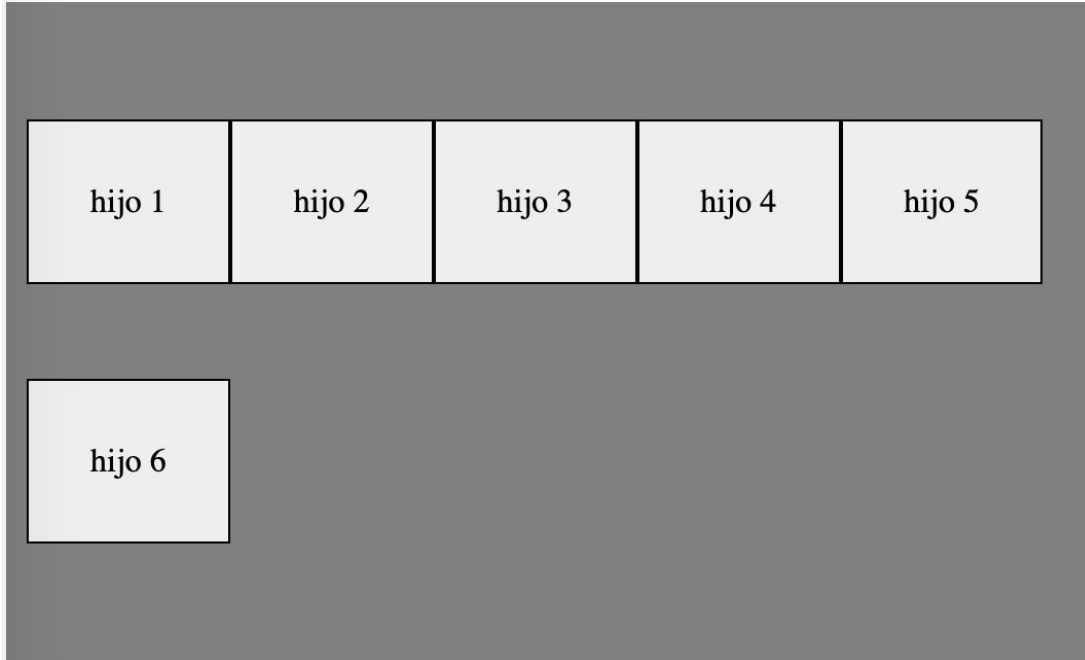


# Align-content: space-around

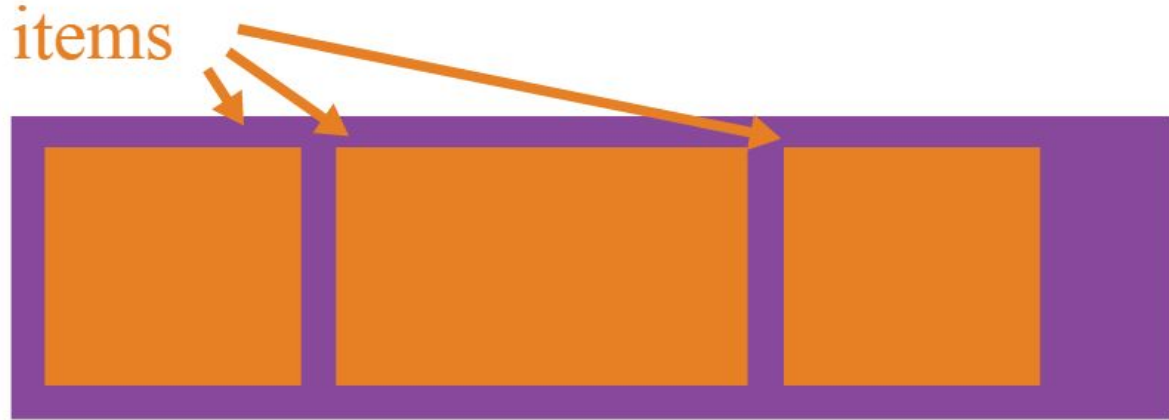




# Align-content: space-evenly



# 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;  
}
```





# 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.



## Ejemplo en vivo

¡Vamos a practicar lo visto!



# Primera pre-entrega de tu Proyecto final



# Primera entrega

**Aclaraciones:** Debe tener el nombre "PreEntrega1+Apellido"

**Sugerencias:** Activar comentarios en el archivo

### Consigna

- ✓ Deberás hacer entrega de tu proyecto donde aparezca: Wireframe/prototipo (wireframe de todas tus páginas para vista mobile y desktop) + HTML (uso de etiquetas contenedoras, etiquetas multimedia y etiquetas de texto) + CSS (con modificadores de texto, colores, listas y box modeling).

### Objetivos

- ✓ **Prototipar la web** para tener una idea clara del resultado al que quieres llegar.
- ✓ **Maquetar la web:** utilizar los tags, en especial los semánticos, para describir la estructura de la web desde el código.
- ✓ **Crear un estilo inicial:** comenzar a darle estilo básico a la web.





# Primera pre-entrega

### Se debe entregar

- ✓ **Prototipo de la web:** versión que muestre cómo se verá el sitio cuando esté productivo.  
Crear una estructura donde se organicen los elementos que van a estar en la web. El nivel de detalle no es importante, sino más bien las posiciones que los elementos van a tener y su tamaño aproximado.  
**Formato:** Archivo PDF o de Imagen

- ✓ **Estructura inicial de la web en HTML:** llevar el contenido a la estructura HTML haciendo uso de los tags que corresponden para el contenido a insertar.  
Incluyendo: **etiquetas semánticas** (tags semánticos de HTML5), **contenido** (agregar etiquetas que van a servir para denotar dónde va a haber contenido como imágenes, párrafos y titulares) y **páginas** (incluir las secciones del sitio ya maquetadas con la estructura propia de cada página)  
**Formato:** Archivos HTML



# Primera pre-entrega

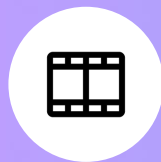
- ✓ **Estilo inicial de la web en CSS:** comenzar a darle estilo básico a una página de su sitio web (ej: index.html) aplicando con CSS las propiedades vistas hasta el momento para modificar textos, encabezados, img, colores, background y box modeling.

El documento debe estar linkeado en las páginas del proyecto.

**Formato:** Archivo CSS

\*Flexbox **no debe incluirse en esta entrega.**

¿Preguntas?



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



MATERIAL AMPLIADO

# Recursos multimedia

- ✓ [Imágenes de relleno](#) | placekitten
- ✓ [Imágenes de relleno](#) | placedog.net
- ✓ [The CSS Box Model](#) | CSS-TRICKS

Disponible en nuestro repositorio.

**CODERHOUSE**

# Resumen de la clase hoy

- ✓ Cajas y sus propiedades.
- ✓ Modelación de la página web.
- ✓ Posiciones de un elemento.

**Opina y valora**  
esta clase

**Muchas gracias.**



**#DemocratizandoLaEducación**