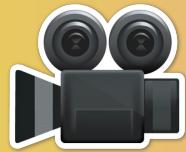


***RECUERDA PONER A GRABAR LA
CLASE***





¿DUDAS DEL ON-BOARDING?

[MIRALO AQUI](#)



Clase 05. DESARROLLO WEB

FLEXBOX



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:

Clase 4

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.

Espacio exterior

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.

GLOSARIO:

Clase 4

Display: se encarga de definir cómo se ve un elemento HTML.

Inline-block: esta propiedad permite tomar lo mejor de ambos grupos. Brinda la posibilidad de tener “padding” y “margin” hacia arriba y abajo.

Float: la flotación consiste en mover un elemento hacia la derecha o izquierda de su línea, y todo lo que viene después se acomodará en el “hueco” que queda vacío.

Clear: esta propiedad permite modificar el comportamiento por defecto del posicionamiento flotante, para forzar a un elemento a mostrarse debajo de cualquier caja flotante.

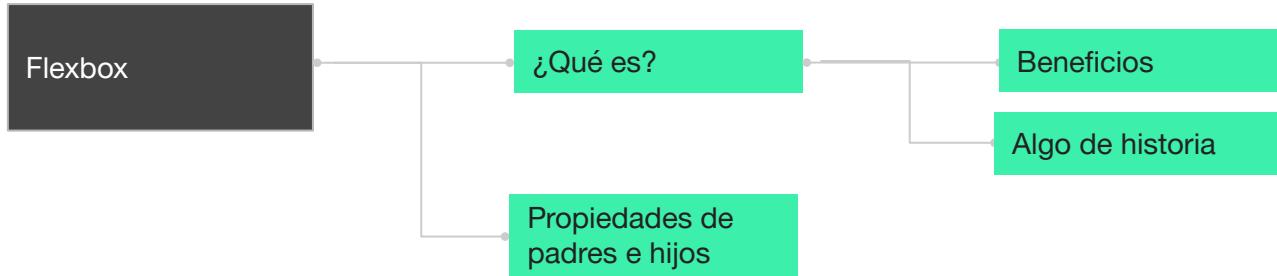
Position: una propiedad CSS pensada para ubicar un elemento, con una libertad muy flexible.

Propiedad z-index: entra en juego cuando dos elementos que tienen position se superponen. Acepta como valor un número (sin ninguna unidad, ni px, ni cm, ni nada); a valor más alto, se mostrará por encima de los demás elementos.

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 5

¡Para
recordar!



CRONOGRAMA DEL CURSO

Clase 4



CSS + Box Modeling



PRÁCTICAS DE LO
VISTO EN CLASE



ESTILOS



Incluir CSS en nuestro
Proyecto

Clase 5



Flexbox



PRÁCTICAS DE LO
VISTO EN CLASE



PRIMERA ENTREGA
DEL PROYECTO FINAL

Clase 6



Grids



PRÁCTICAS DE LO
VISTO EN CLASE



GUIÓN DE LA CLASE

Accede al material complementario [aquí](#).



FLEXBOX

CODERHOUSE

¿QUÉ ES FLEXBOX?

¿QUÉ ES FLEXBOX?

- 👉 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 sólo 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...

td | 130 x 53

Upload, tag and share your videos worldwide!

[Sign Up](#) | [Log In](#) | [Help](#)

[Upload Videos](#) // [Browse Videos](#)

[Home](#) | [My Videos](#) | [My Favorites](#) | [My Profile](#)

Upload
Quickly upload and tag videos in almost any video format.

Watch
Instantly find and watch 1000's of fast streaming videos.

Share
Easily share your videos with family, friends, or co-workers.

Sign up for your free account!

Recent Tags:

[guitar](#) : [francis](#) : [video](#) : [xts912](#) :
[belgium](#) : [man](#) : [driving](#) : [fall](#) :
[woman](#) : [isl](#) : [flash](#) : [iran](#) :
[orange](#) : [scary](#) : [soccer](#) :
[homestead](#) : [live](#) : [CRAZY](#) :
[fight](#) : [cute](#) : [kitten](#) :
[kittens](#) : [cute](#) : [kitten](#) :

Today's Featured Videos

Tiffany
Is a total dork

[Tags](#) // [Tiffany](#) : [dork](#) : [funny](#) : [random](#) .

[See More Videos](#)

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



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

The screenshot shows the YouTube homepage from February 2010. A green callout box highlights the class selector "#homepage-main-content.homepage-non-interactive.homepage-new-skin". To the right, the browser's developer tools are open, showing the CSS for the "Homepage" skin. The "float" property is highlighted in red in the CSS for the main content container. The page displays several video thumbnails and titles, such as "Interview with Obama" and various user-submitted questions.

```
#homepage-main-content.homepage-non-interactive.homepage-new-skin { width: 625px; }
```

```
.homepage-new-skin { float: left; clear: left; width: 640px; margin-top: 5px }
```

Thumbnail	Title	Length	Uploader	Published	Views	Rating	Comments
	Why are banks reluctant to modif...	35:47	resi...	5 days ago	443 views	★★★★★	0334231
	State of the Union Question	0:25		5 days ago	3,337 views	★★★★★	energyaction
	President Obama - How will you h...	0:17	ENOUGHproject	5 days ago	1,494 views	★★★★★	

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



En la actualidad... se aplica **Flexbox**:

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

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;
```

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

EJEMPLOS

Páginas con Flexbox:



YouTube AR

Buscar

ACCEDER

Principal

Tendencias

Suscripciones

Biblioteca

Historial

Pasión Por La Roja ✓
676,012 vistas • hace 5 años

Todo Noticias ✓
110 vistas • hace 17 minutos

Conciertos desde casa Destacado
No te pierdas los conciertos más recientes de tus artistas favoritos

Priceless Experiences at Home | Camila Cabello
Camila Cabello 1.2 M de vistas • Transmitido hace 2 meses

Coke Studio Sessions: Katy Perry
Katy Perry 728,486 vistas • Transmitido hace 1 mes

Elements Console » 1 1 1 1

yt-dBrowse "disavow-upgrade hidden"
</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-...
... ytd-two-column-browse-results-renderer ...

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

yt-dBrowse-browse-
results-renderer {
 (index):41998
 margin-bottom: 16px;
 width: 100%;
 display: flex;
 flex-direction: column-reverse;
 justify-content: flex-end;
}
Inherited from ytd-page-man...
#page-manager.ytd-app {
 (index):57526
 -ytd-toolbar-offset: var(--ytd-masthead-height, var(--ytd-toolbar-height));
 overflow-x: auto;
 margin-top: var(--ytd-masthead-height, var(--ytd-toolbar-height));
 display: -ms-flexbox;

Filter Show all
display
flex
flex-direction
column-reverse
font-family
Roboto, Arial,

Páginas con Flexbox:



A screenshot of a YouTube homepage on a desktop browser. The interface includes a top navigation bar with a search bar and a 'ACCEDER' button. On the left, there's a sidebar with links for 'Principal', 'Tendencias', 'Suscripciones', 'Biblioteca', and 'Historial'. The main content area is titled 'Recomendados' and shows a grid of video thumbnails. Some visible titles include 'Abel Pintos - Campaña Codo a Codo', 'No sabe nadar', 'Obra pública: Testimonio desbarata acusación contra Cristina', 'THE SPONGEBOB MOVIE: Sponge on the Run Trailer', 'Los Palmeras Desde Casa - Enganchado del Recuerdo', and 'CHILI con carne'.

Desktop

A screenshot of a YouTube homepage on a tablet device. The layout is similar to the desktop version, showing a 'Recomendados' section with various video thumbnails. Some visible titles include 'Doblaje Argento - PON ESA COSA HORROROSA AHÍ', 'Jesse & Joy - Lo Nuestro Vale Más (Video Oficial)', 'Zorrito Amigable Un Poco De Todo', 'Casados Con Hijos 2020 - Detrás de Cámaras', 'Los Palmeras Desde Casa - Enganchado del Recuerdo', and 'MINIONS 2: The Rise of Gru Trailer (2021)'.

Tablet

A screenshot of a YouTube homepage on a mobile device. The interface is adapted for a smaller screen, showing a 'Recomendados' section with video thumbnails. Some visible titles include 'Los Palmeras Desde Casa - Enganchado del Recuerdo', 'Doblaje Argento - PON ESA COSA HORROROSA AHÍ', 'Zorrito Amigable Un Poco De Todo', 'Casados Con Hijos 2020 - Detrás de Cámaras', 'Los Palmeras Desde Casa - Enganchado del Recuerdo', and 'MINIONS 2: The Rise of Gru Trailer (2021)'.

Mobile

Páginas con Flexbox:

The screenshot displays the Amazon homepage with three main grid-based sections:

- Compra por Categoría**: Features categories like Computadoras, Videojuegos, Bebé, and Juguetes y Juegos.
- AmazonBasics**: Shows a product image of a USB cable.
- Electrónicos**: Shows a product image of a laptop keyboard and a Polaroid camera.

A developer tools sidebar on the right shows the CSS for the 'Electrónicos' section, specifically targeting the '.fluid-card' class with a red box around it. The CSS includes:

```
.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:



We ship over 45 million products around the world

Shop by Category

- Computers & Accessories
- Video Games

AmazonBasics

Electronics

Sign in for the best experience

Sign in securely

Desktop

We are delivering to your region with limited shipping options. Please expect extended delivery time. [Learn more.](#)

Shop Computers & Accessories

Shop by Category

- Computers & Accessories
- Video Games

AmazonBasics

Electronics

Sign in for the best experience

Sign in securely

Computers & Accessories

Beauty picks

Tablet

We ship 45 million products around the world

Sign in for the best experience

Sign in

Create an account

Deal of the Day

Inflatable Mattress

Mobile

Páginas con Flexbox:



The screenshot shows the browser's developer tools with the "Styles" tab selected, displaying the CSS properties for the element with class "css-1dbjc4n r-13qz1uu". The properties listed include:

```
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;
```

Below this, the "Computed" tab shows the user agent stylesheet with the "display: block;" rule.

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?

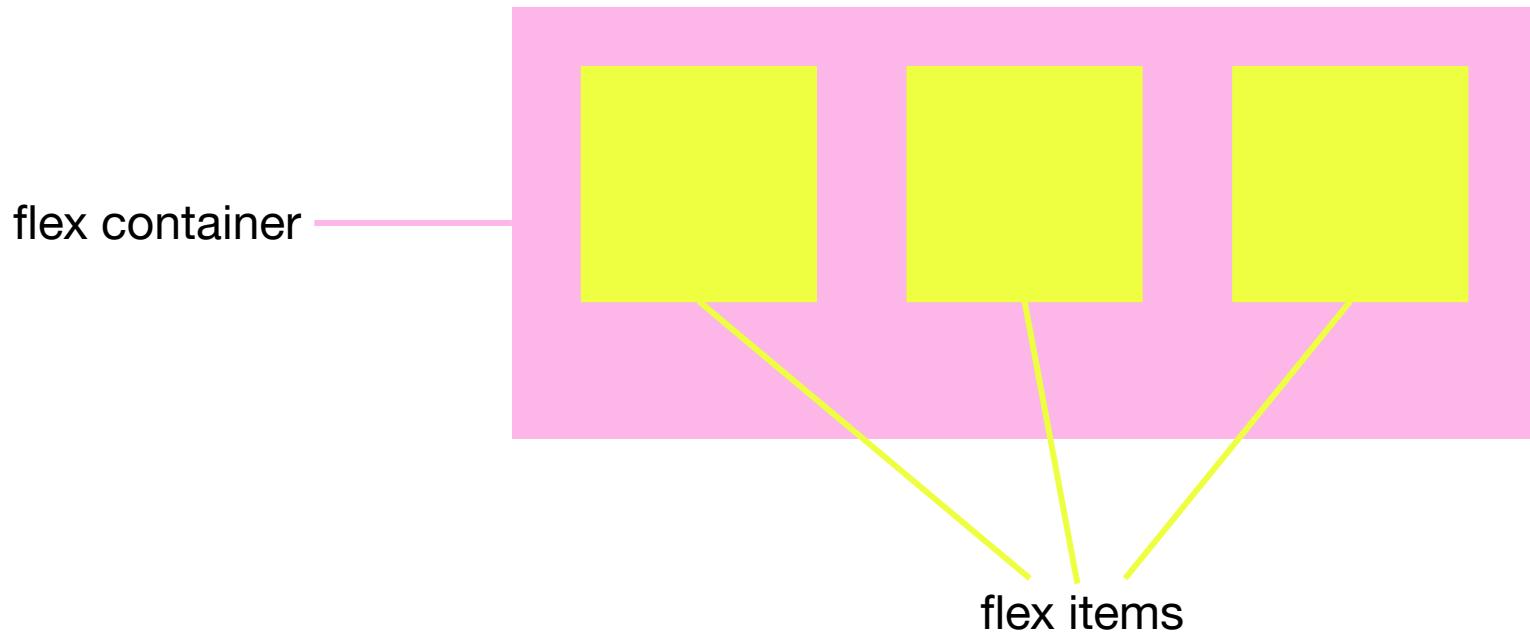
flex-flow Abreviación de propiedades.

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?



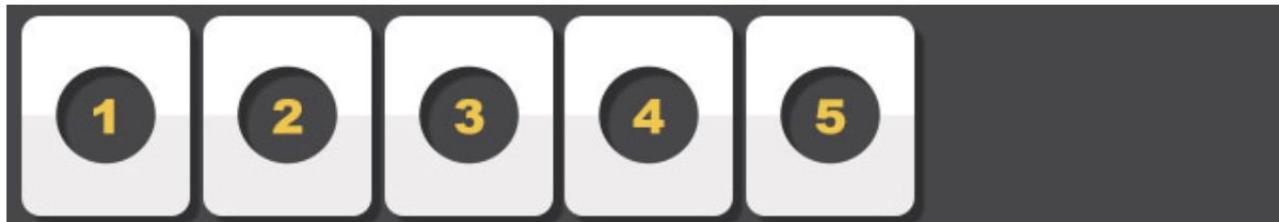
👉 Lo primero que debemos hacer es establecer la propiedad display con el valor flex en el elemento padre.

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

FLEX-DIRECTION: 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 */  
}
```



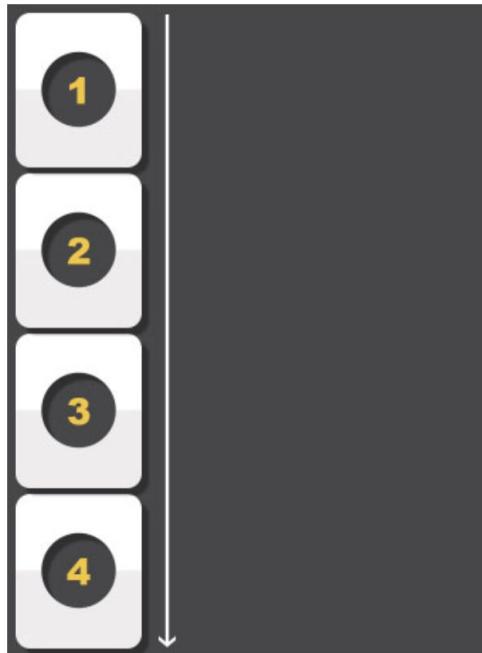
FLEX-DIRECTION: ROW-REVERSE

👉 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;  
}
```



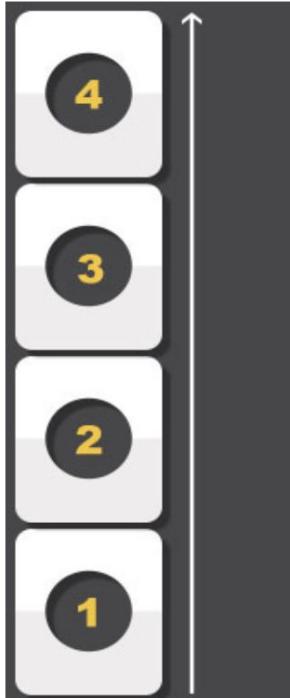
FLEX-DIRECTION: COLUMN



👉 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-REVERSE



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;  
}
```



row-reverse



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

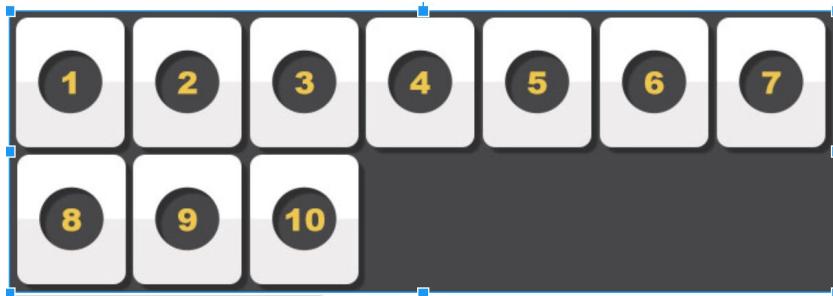


```
.padre-flex {  
    display: flex;  
    flex-wrap: nowrap;  
}
```



FLEX-WRAP: WRAP

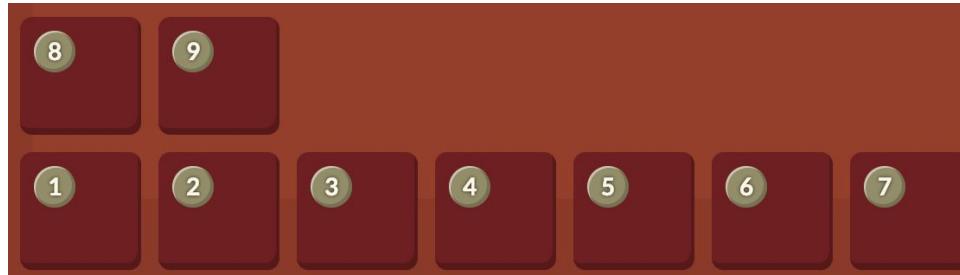
Los *flex items (hijos)* pueden romper la línea del eje horizontal, si les es necesario para conservar las características de sus dimensiones. Esto es de izquierda a derecha, y de arriba a abajo.



```
.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;  
}
```



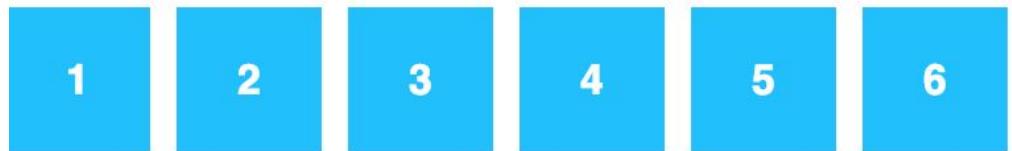
nowrap



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;  
}
```

JUSTIFY-CONTENT

- *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.
- **Los siguientes ejemplos parten de la base del contenedor en “row”.**



JUSTIFY-CONTENT

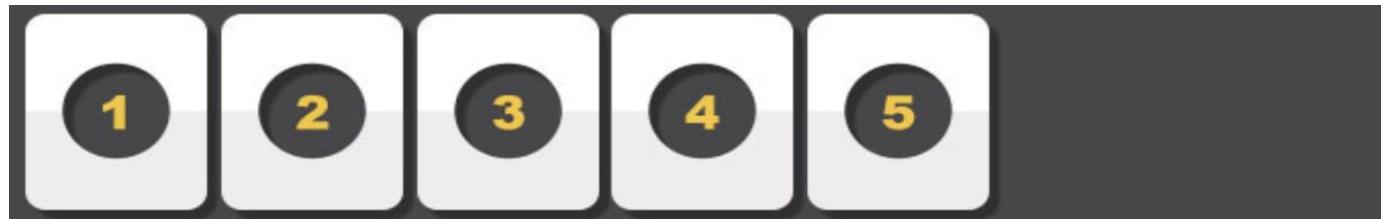
Los siguientes ejemplos parten de la base del contenedor en “row”:

```
.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.

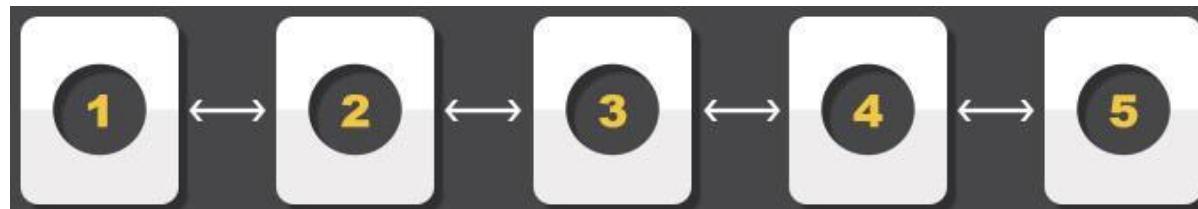
```
.padre-flex {  
    display: flex;  flex-direction: row;  
    justify-content: center;  
}
```



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.

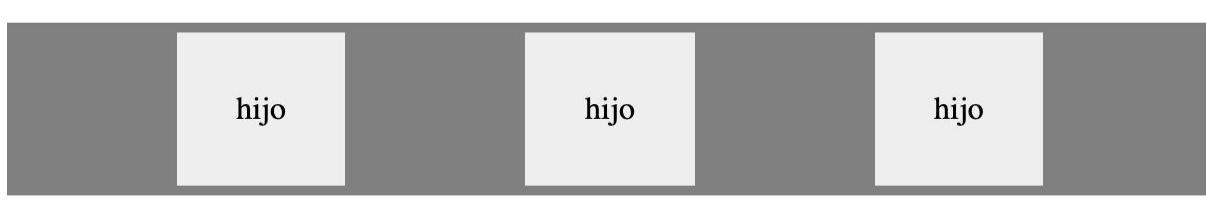
```
.padre-flex {  
    display: flex;  flex-direction: row;  
    justify-content: space-around;  
}
```

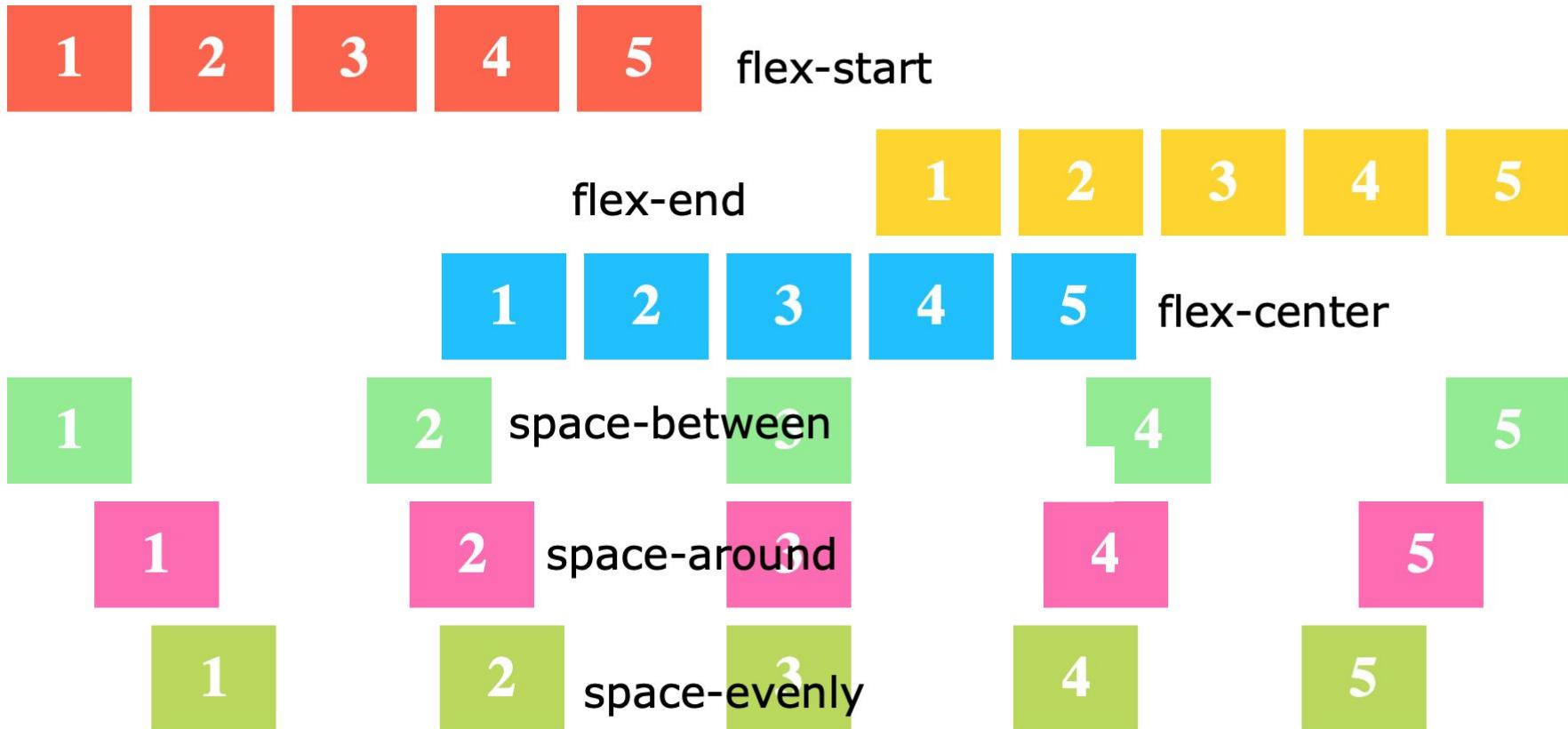


JUSTIFY-CONTENT: SPACE-EVENLY

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



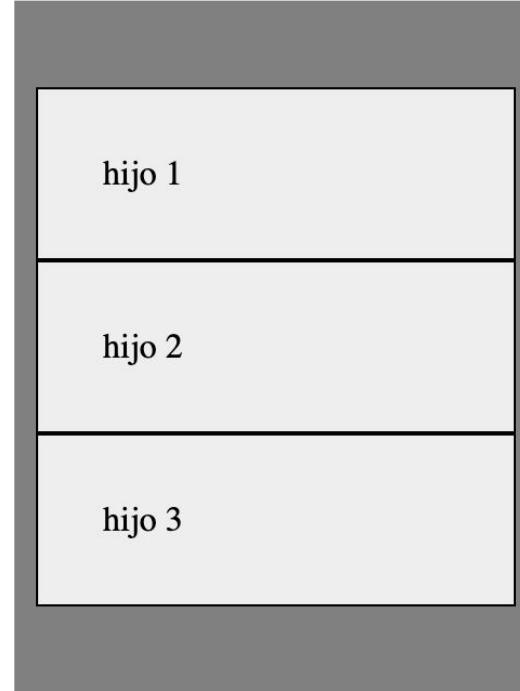
```
.padre-flex {  
    display: flex;  
    flex-direction: column;  
    justify-content: flex-end;  
    height: 300px;  
}
```



JUSTIFY-CONTENT: CENTER



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



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;  
}
```



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

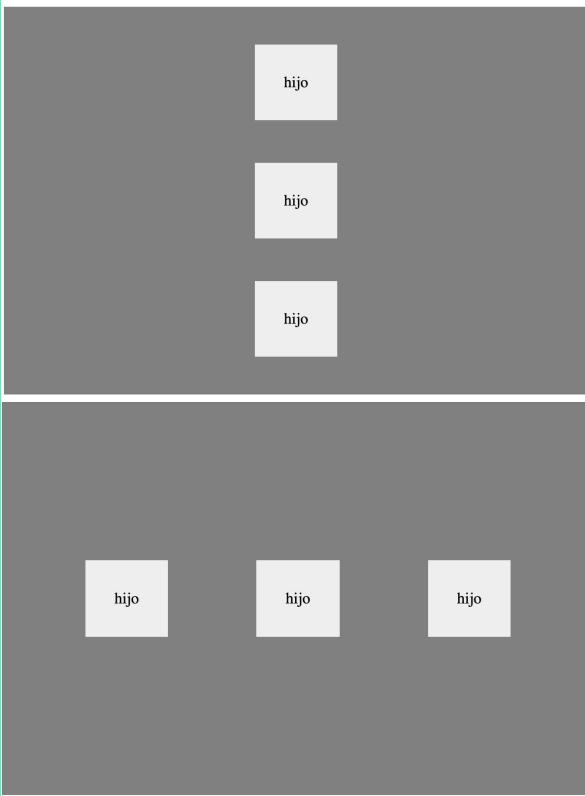


```
.padre-flex {  
    display: flex;  
    flex-direction: column;  
    justify-content: space-evenly;  
    height: 300px;  
}
```



ALIGN-ITEMS

👉 Alinear los elementos verticales de forma horizontal.
(flex-direction: column)

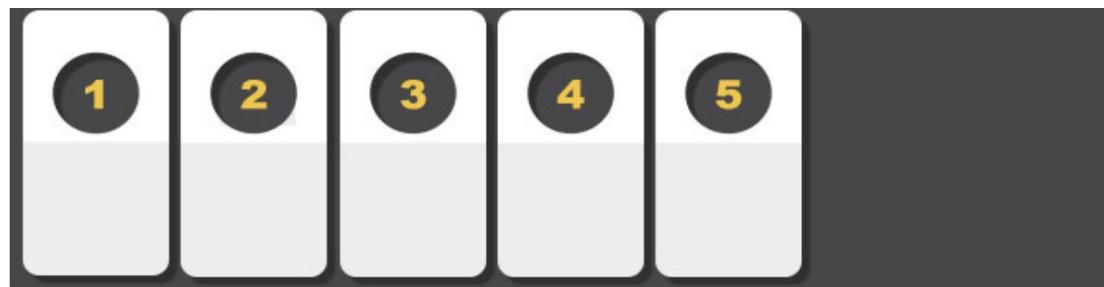


👉 Alinear los elementos horizontales de forma vertical.
(flex-direction: row)

ALIGN-ITEMS: STRETCH

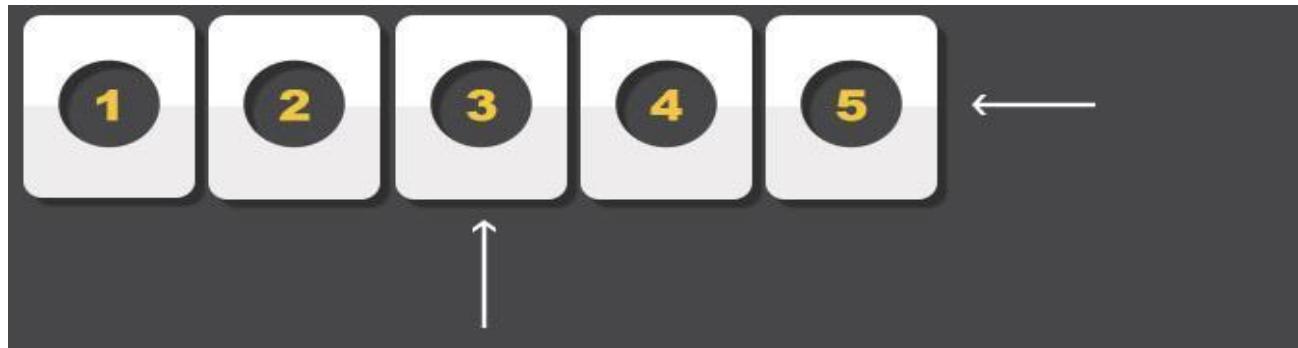
Tratará de llenar toda la altura (o anchura) del contenedor, siempre y cuando los *hijos* no tengan propiedades de dimensión definidas.

```
.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 */  
}
```



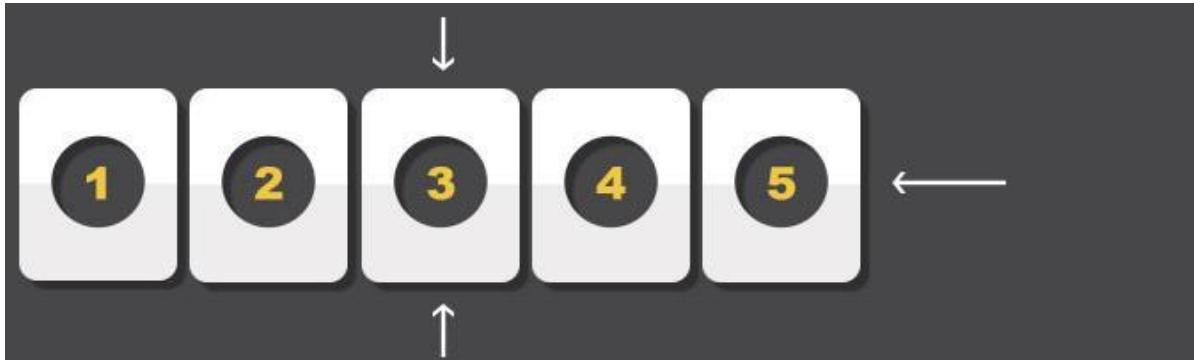
ALIGN-ITEMS: FLEX-END

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



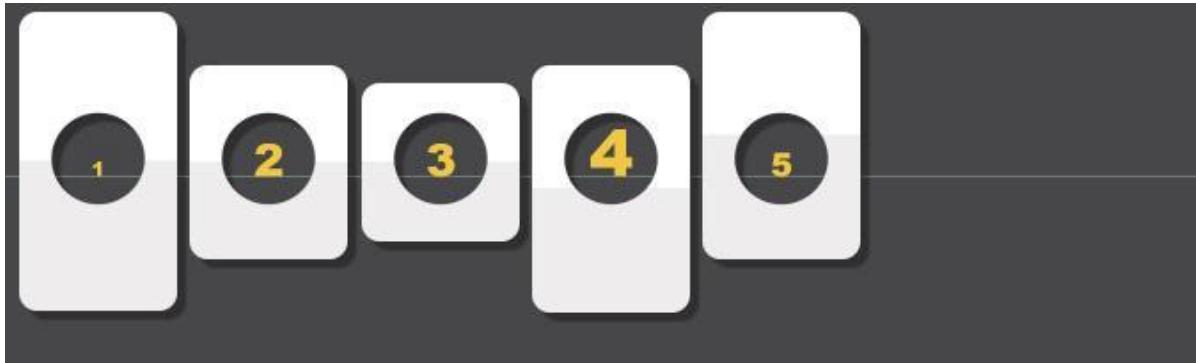
ALIGN-ITEMS: CENTER

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



ALIGN-ITEMS: *BASELINE*

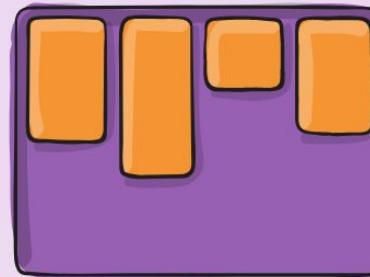
```
.padre-flex {  
    display: flex; flex-direction: row;  
    align-items: baseline; /* predeterminado */  
}
```



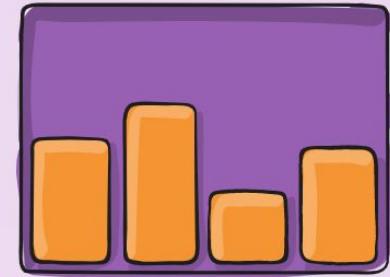
ALIGN-ITEMS

RESUMEN

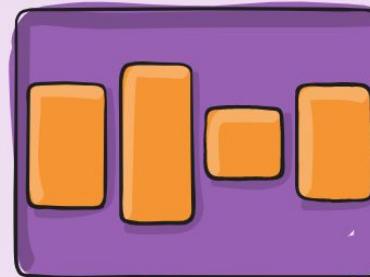
flex-start



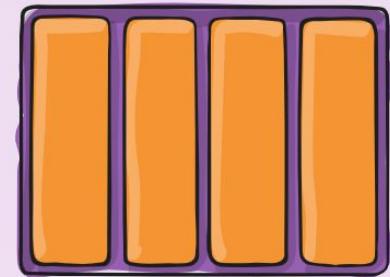
flex-end



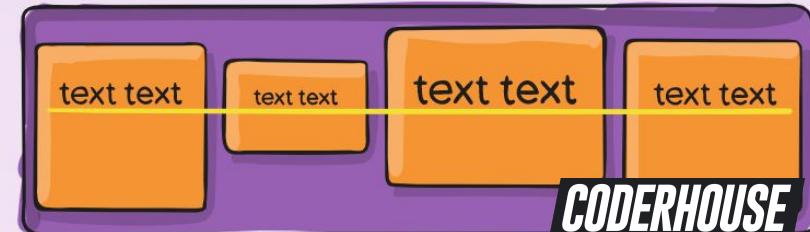
center



stretch



baseline



ALIGN-CONTENT

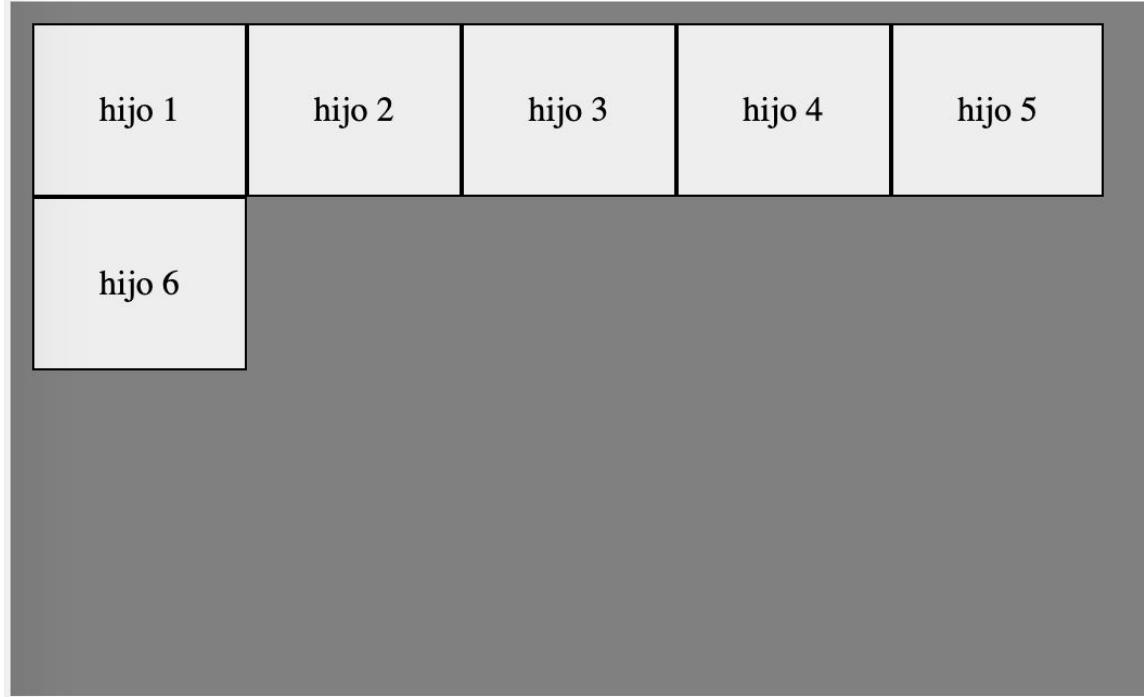
- 👉 Esta propiedad sólo tiene efecto cuando el contenedor flexible tiene varias líneas de *flex items (hijos)*.
- 👉 Si se colocan en una sola línea, esta propiedad no tiene ningún efecto sobre el diseño.
- 👉 Para poder aplicarlo se necesita tener el atributo *flex-wrap*, que permita verificar los ejes horizontales.

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

ALIGN-CONTENT: STRETCH



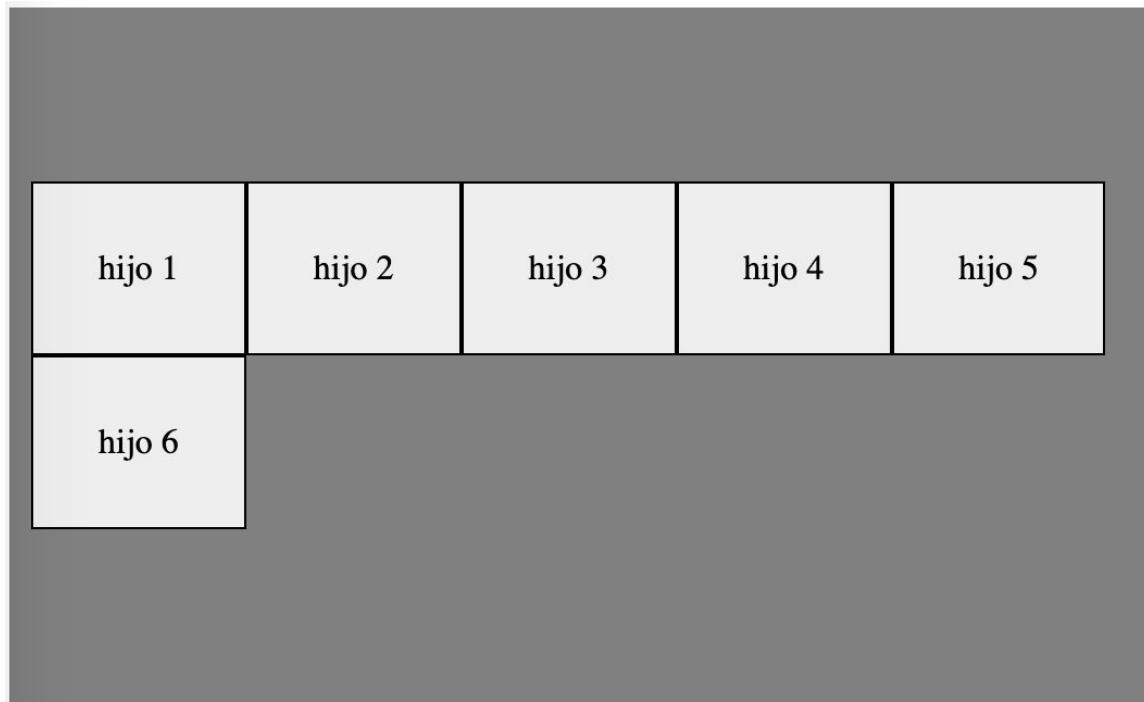
ALIGN-CONTENT: FLEX-START



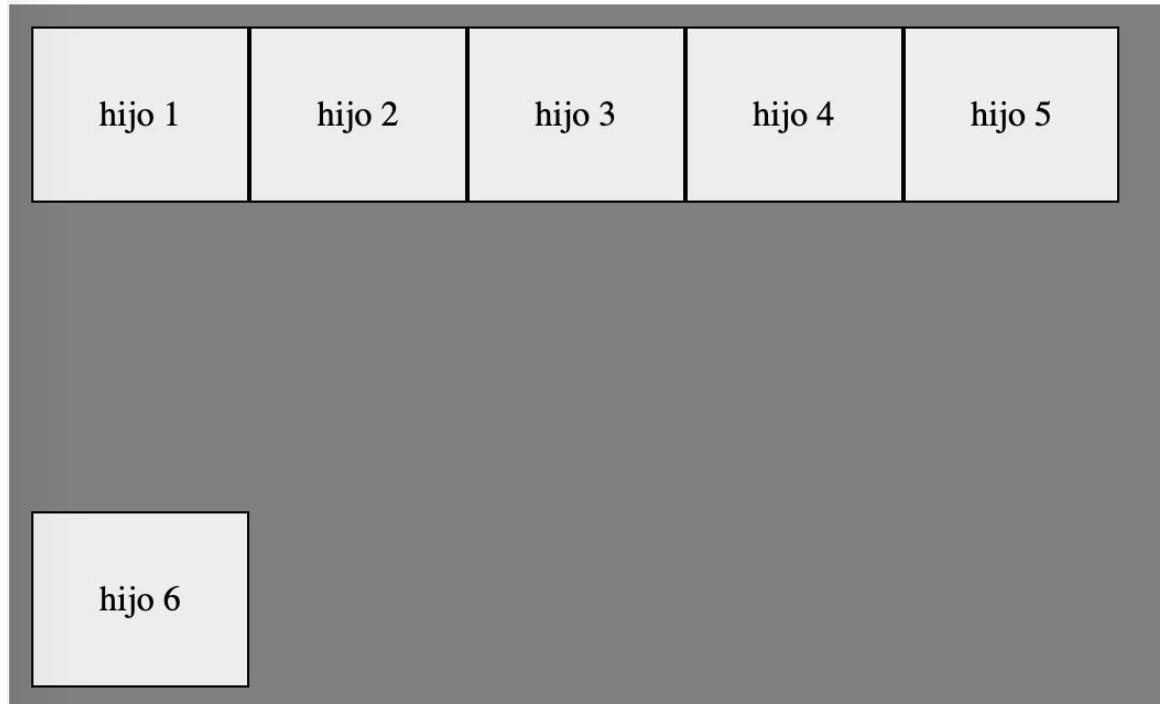
ALIGN-CONTENT: FLEX-END



ALIGN-CONTENT: CENTER



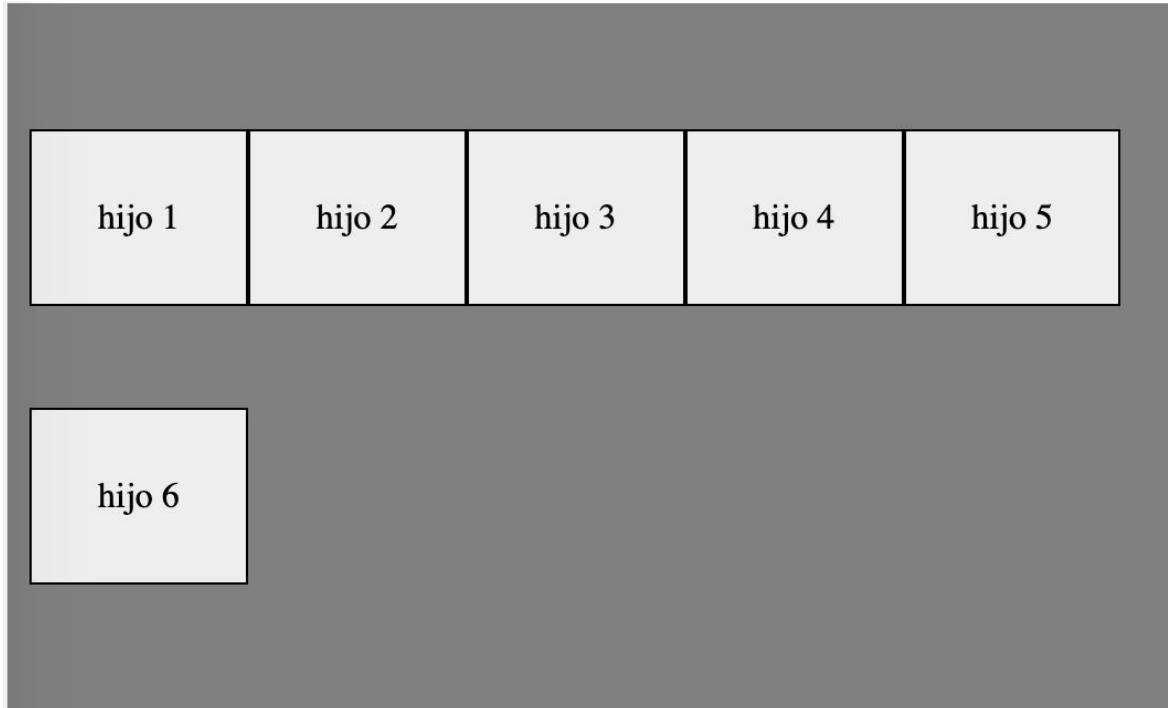
ALIGN-CONTENT: SPACE-BETWEEN



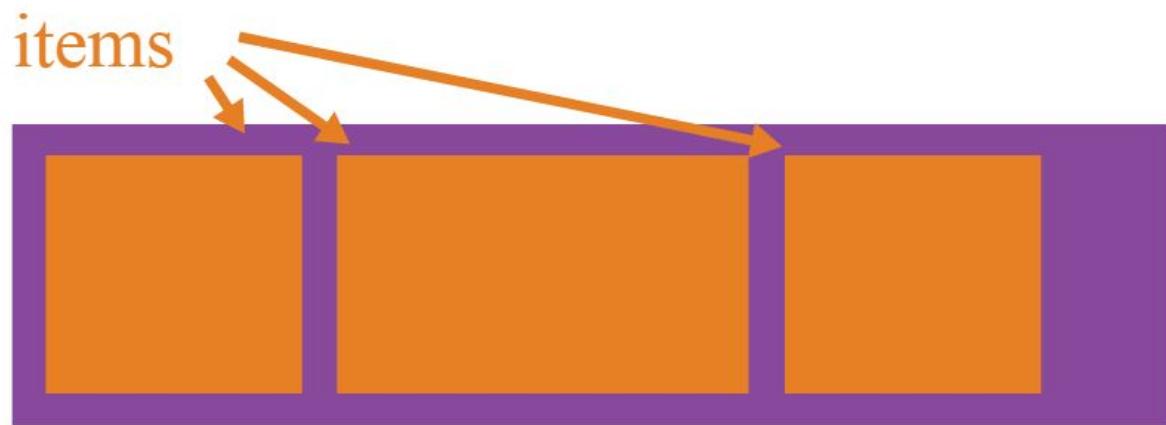
ALIGN-CONTENT: SPACE-AROUND



ALIGN-CONTENT: SPACE-EVENLY



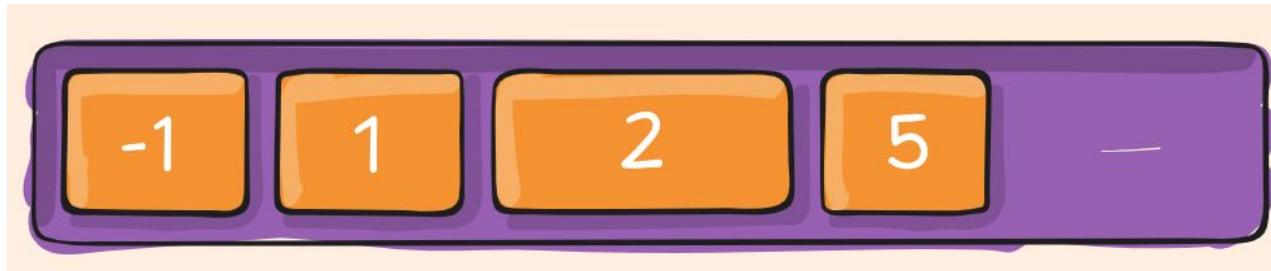
PROPIEDADES PARA LOS FLEX ITEMS (HIJOS)



ORDER

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

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





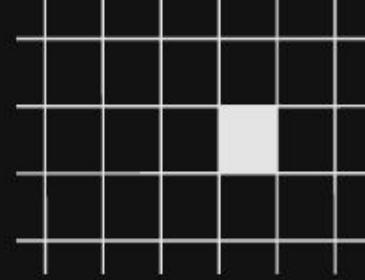
BREAK

¡5/10 MINUTOS Y VOLVEMOS!



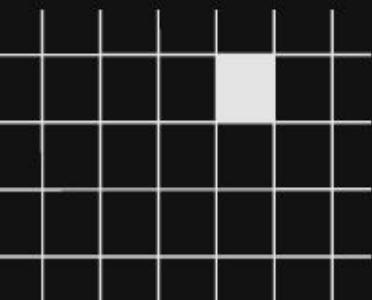
conectados
con las *oportunidades*

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.



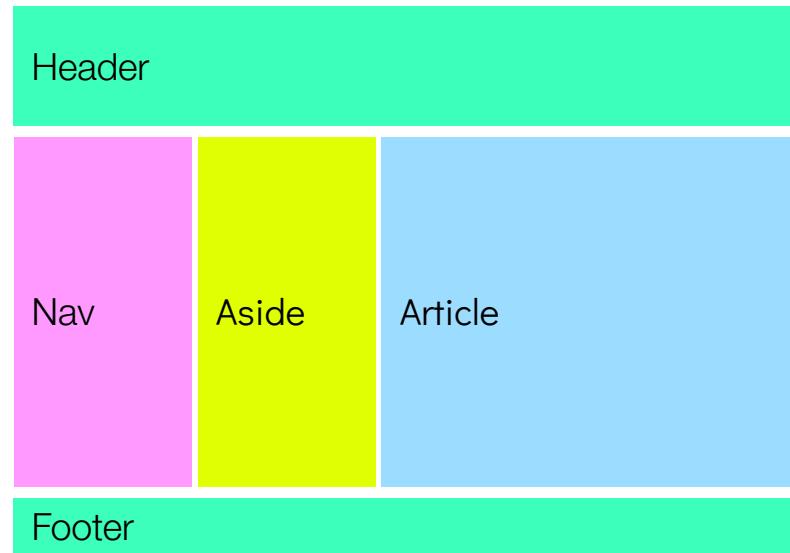
Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO!

PÁGINA CON FLEXBOX

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



PÁGINA CON FLEXBOX

CSS

```
body {  
    display: flex;  
    height: 600px;  
    flex-direction: column;  
    padding: 1em;}  
  
#main{  
    display: flex;  
    flex: 1;}  
  
article, nav, aside{  
    background-color: pink;}  
  
header, footer{  
    background-color: grey;  
    margin: 1px;}  
  
article, nav, aside, header,  
footer{  
    flex: 1;  
    padding: 1rem;  
    margin: 1px;}
```

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;  
    height: 600px;  
    flex-direction: column;  
    padding: 1em;}  
  
#main{  
    display: flex;  
    flex: 1;}  
  
article, nav, aside{  
    background-color: pink;}  
  
header, footer{  
    background-color: grey;  
    margin: 1px;}  
  
article, nav, aside, header,  
footer{  
    flex: 1;  
    padding: 1rem;  
    margin: 1px;}
```

HTML

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



PRIMERA ENTREGA DEL PROYECTO FINAL

PRIMERA ENTREGA DEL PROYECTO FINAL

Formato: Aclarar tipo de archivo en que se espera el desafío. Debe tener el nombre “Idea+Apellido”.

Sugerencia: Activar comentarios en el archivo.

Proyecto
Final



1

>>Objetivos Generales:

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

>>Objetivos Específicos:

1. **Prototipar la web** para tener una idea clara del resultado al que quiere llegar.
2. **Maquetar la web:** El estudiante deberá utilizar los tags, en especial los semánticos, para desde el código describir la estructura de la web.
3. **Crear un estilo inicial:** El estudiante deberá comenzar a darle estilo básico a su web definiendo colores. El documento debe estar linkeado en las páginas del proyecto.

PRIMERA ENTREGA DEL PROYECTO FINAL

Formato: aclarar tipo de archivo en que se espera el desafío. Debe tener el nombre “Idea+Apellido”.

Sugerencia: activar comentarios en el archivo.

Proyecto
Final



>>Se debe entregar:

- Wireframe/prototipo completo.
- Estructura de la web en HTML: página principal (index.html) + cada una de las páginas necesarias dependiendo de cada proyecto y los enlaces entre sí.
- Estilo básico: añadir estilo utilizando las propiedades y valores CSS vistos hasta el momento.
Flexbox **no debe incluirse en esta entrega.**

*¿*PREGUNTAS?

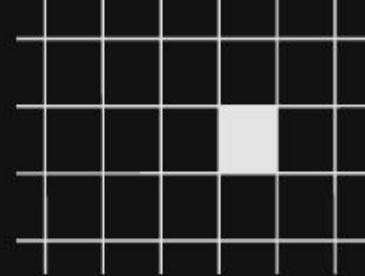




**¿QUIERES SABER MÁS? TE DEJAMOS
MATERIAL AMPLIADO DE LA CLASE**



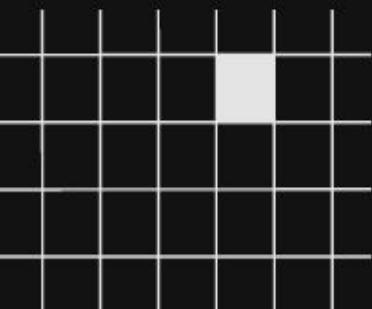
- [Imágenes de relleno](#) | ***placekitten***
- [Imágenes de relleno](#) | ***placedog.net***
- [The CSS Box Model](#) | **CSS-TRICKS**



¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

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





OPINA Y VALORA ESTA CLASE

#DEMOCRATIZANDOLAEDUCACIÓN