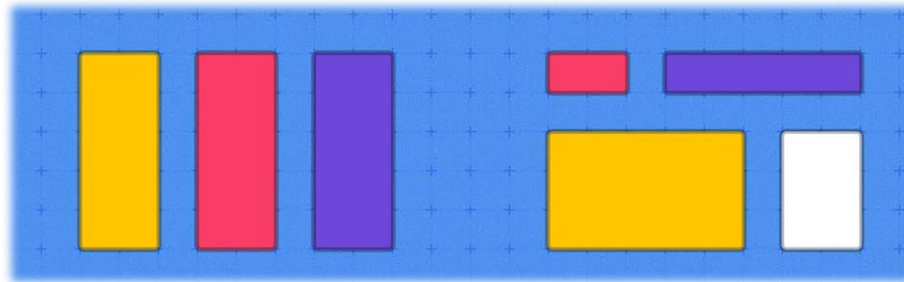


# Diseño de Interfaces Web

## Flexbox



# Flexbox

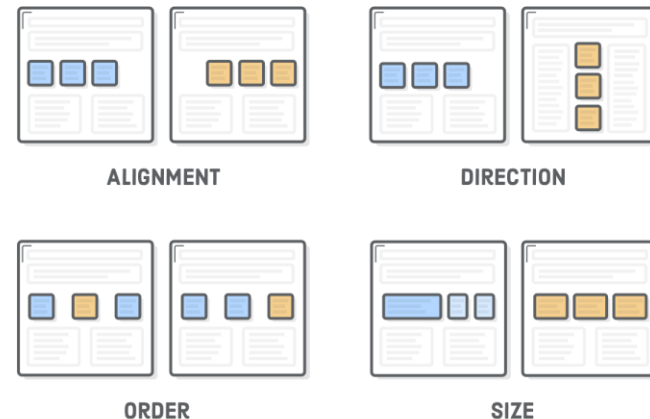
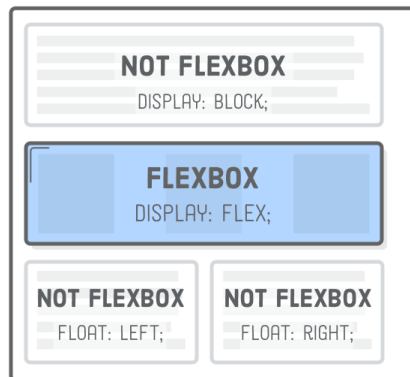


1. [Introducción](#)
2. [Ejes Flexbox](#)
3. [Flex container y flex items](#)
4. [Propiedades flex container](#)
5. [Alineaciones](#)
6. [Propiedades flex items](#)
7. [Espaciado](#)
8. [Anidación de contenedores flex](#)
- \* [Referencias](#)



# 1. Introducción

- ¿Qué es Flexbox?
  - “Modelo de caja flexible”
  - Módulo de CSS3 que proporciona un control específico sobre la alineación, la dirección, el orden y el tamaño de las cajas
  - Trabaja en **una dimensión**
  - “Gran alternativa a los floats tradicionales”



## 2. Ejes Flexbox

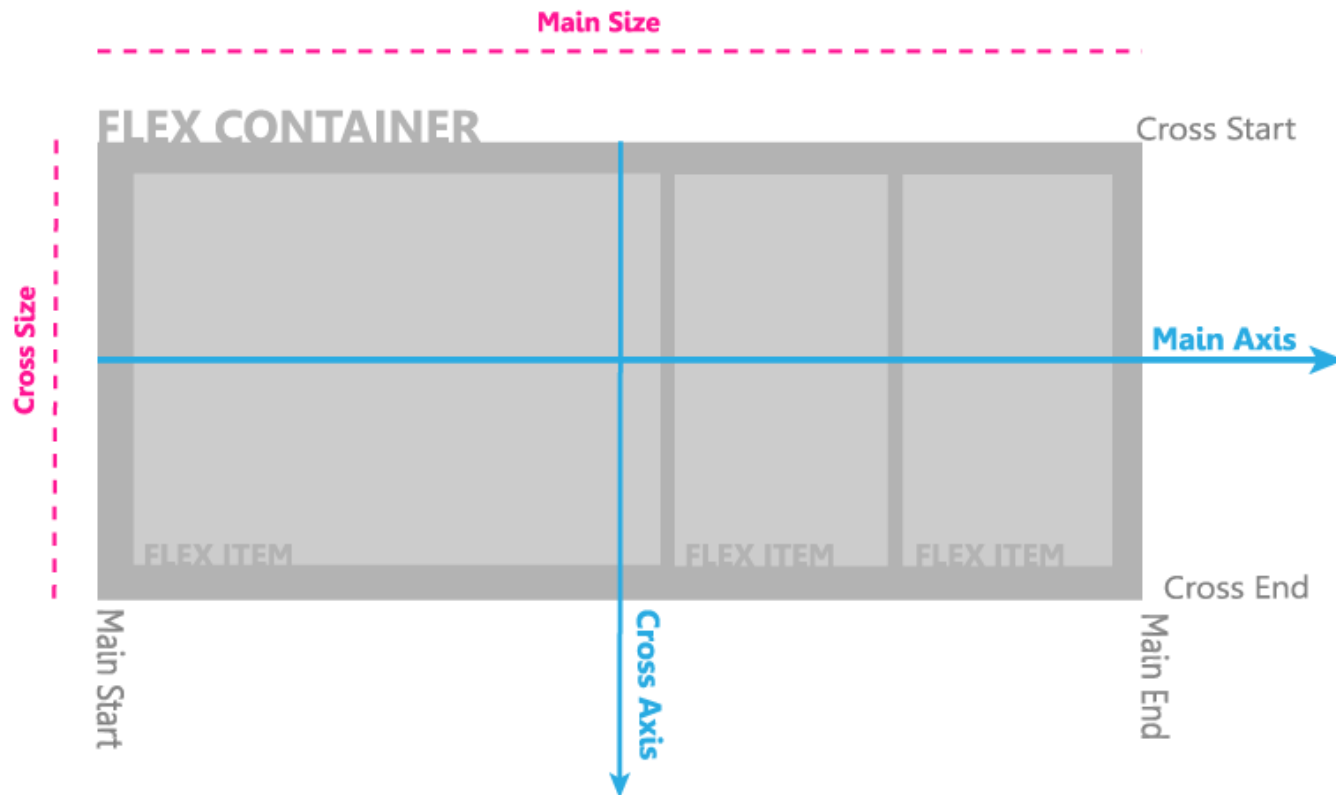


- Ejes en Flexbox
  - En Flexbox tenemos **2 ejes**
    - Eje principal (**main axis**)
    - Eje cruzado (**cross axis**), perpendicular al main axis
  - Por defecto, el eje principal es el eje horizontal y el eje cruzado es el vertical
  - Podemos cambiar este comportamiento con la propiedad ***flex-direction***;

## 2. Ejes Flexbox



- Ejes en Flexbox



# 3. Flex container y flex items



- Contenedores flex y elementos flex
  - A la hora de aplicar propiedades, distinguiremos entre las del **contenedor flex (flex container)** y las de los **elementos flex (flex items)**
  - En el contenedor flex es donde aplicamos **display: flex;** o **display: inline-flex;**
  - Esta propiedad no se hereda, por tanto **solo afecta a los hijos directos del contenedor flex**



“FLEX CONTAINER”



“FLEX ITEMS”

# 4. Propiedades flex container



- Dirección

- **flex-direction**

- flex-direction: **row** / **row-reverse**

- El eje principal es el *horizontal* y el eje cruzado el *vertical*

- flex-direction: **column** / **column-reverse**

- El eje principal es el *vertical* y el eje cruzado el *horizontal*



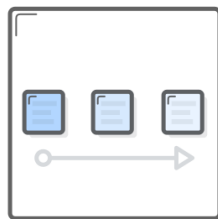
**ROW**

FLEX-DIRECTION: ROW;

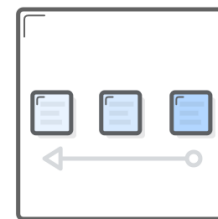


**COLUMN**

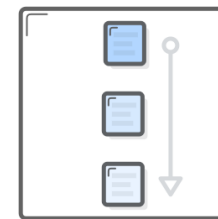
FLEX-DIRECTION: COLUMN;



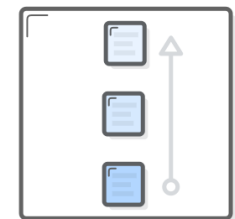
**ROW**



**ROW-REVERSE**



**COLUMN**

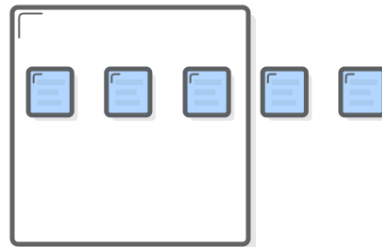


**COLUMN-REVERSE**

# 4. Propiedades flex container

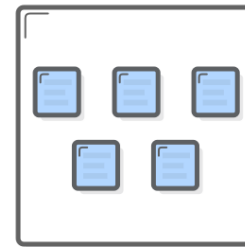


- Wrapping
  - **flex-wrap**
    - wrap / no-wrap / wrap-reverse
      - Define el comportamiento de los flex items cuando el tamaño sumado de todos ellos supera el tamaño del flex container en el eje principal



**NO WRAPPING**

FLEX-WRAP: NOWRAP;



**WITH WRAPPING**

FLEX-WRAP: WRAP;



# 4. Propiedades flex container



- Shorthand flex-flow
  - **flex-flow**
    - Shorthand de las propiedades **flex-direction** y **flex-wrap**
    - Ejemplo: **flex-flow: row wrap;**
    - Por defecto flex-flow: row nowrap;

# 5. Alineaciones



- Alineaciones en el eje principal
  - **justify-content**
    - Permite alinear los elementos en el **eje principal**
    - Valores: *flex-start* (por defecto) / *flex-end* / *center* / *space-between* / *space-around* / *space-evenly*



JUSTIFY-CONTENT



FLEX-START



CENTER



FLEX-END



SPACE-AROUND



SPACE-BETWEEN

# 5. Alineaciones



- Alineaciones en el eje cruzado
  - **align-items**
    - Permite alinear los elementos en el **eje cruzado**
    - Valores: *stretch* (por defecto) / *flex-start* / *center* / *flex-end* / *baseline*



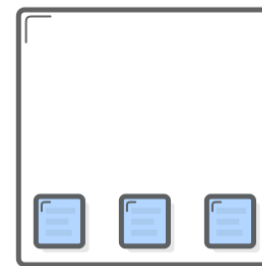
**ALIGN-ITEMS**



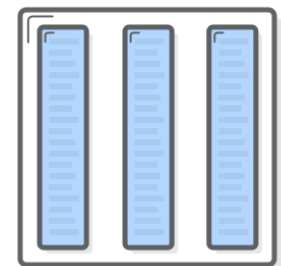
**FLEX-START**



**CENTER**



**FLEX-END**



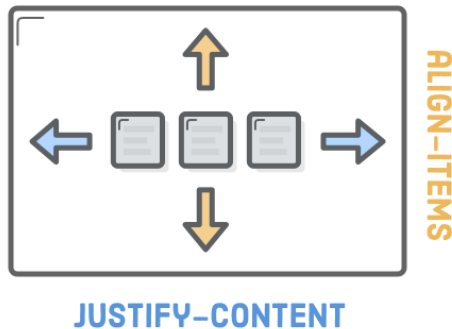
**STRETCH**

# 5. Alineaciones

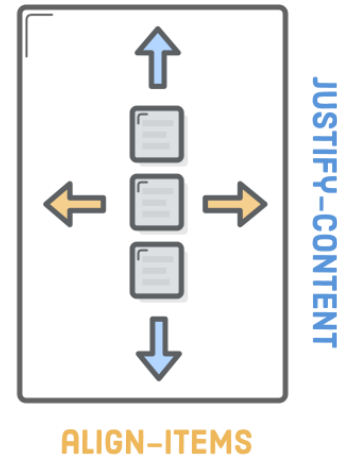


- Consideraciones sobre la alineación
  - Cambio de eje según valor de flex-direction

FLEX-DIRECTION: ROW;



FLEX-DIRECTION: COLUMN;



# 5. Alineaciones



- Alineaciones de elementos concretos
  - **align-self**
    - Permite alinear **elementos flex concretos** con respecto a la alineación general en el **eje cruzado**
    - Mismos valores que align-items:  
*stretch* (por defecto) / *flex-start* / *center* / *flex-end* / *baseline*
  - Flexbox no proporciona un align-self para el **eje principal**, pero podemos aplicarle una alineación a un elemento concreto en este eje con **margin** y el valor **auto** (usaremos *margin-left*, *margin-right*, *margin-top*, *margin-bottom* según el caso)

# 5. Alineaciones



- Alineaciones de más de una fila de elementos flex
  - **align-content**
    - Determina cómo las **líneas flex** (no los items) se colocan sobre el eje cruzado
    - Solo se puede utilizar cuando tenemos **más de una línea flex** y hay espacio sobrante (y con propiedad **flex-wrap: wrap**; en el contenedor flex)
    - Valores: *stretch* (por defecto) / *flex-start* / *center* / *flex-end* / *space-around* / *space-between*

# 6. Propiedades flex items



- Tamaño de los elementos flex
  - **flex-grow**
    - “Factor de crecimiento” de los elementos flex
    - Permite a una caja crecer cuando sobra espacio dentro del espacio flex
    - Si **flex-grow: 1**; la caja crece cuando hay espacio disponible
    - Se puede aplicar a uno o más elementos flex en diferentes proporciones
    - Por defecto los elementos flex tienen **flex-grow: 0**;
    - Solo se aplica en el eje principal

# 6. Propiedades flex items



- Tamaño de los elementos flex
  - **flex-shrink**
    - “Factor de estrechamiento” de los elementos flex
    - Por defecto todos los elementos flex tienen **flex-shrink: 1**;
    - Esto significa que cuando no hay sitio suficiente en el espacio flex, los elementos se reducen proporcionalmente en base a 1
    - Si aplicamos un valor más alto de flex-shrink a un elemento concreto, se reducirá a un ritmo mayor que otros elementos con un valor menor de flex-shrink
    - Si **flex-shrink: 0**; el elemento no reduce
    - Solo se aplica en el eje principal



# 6. Propiedades flex items

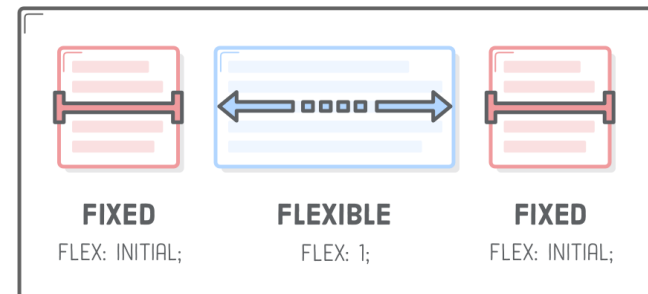


- Tamaño de los elementos flex
  - **flex-basis**
    - Define el **tamaño base** (tamaño inicial) de un ítem en un contenedor flex antes de verse afectado por flex-grow o flex-shrink
    - Un elemento con un valor en la propiedad **flex-basis** intentará ajustarse a ese tamaño dentro del eje principal
    - Funciona de manera similar a *width* pero teniendo en cuenta que es un **valor mínimo** que puede crecer según el contenido del elemento y que **se aplica al eje principal** según *flex-direction*
    - flex-basis predomina sobre *width* o *height*
    - Valor por defecto: **flex-basis: auto;**
      - En este caso el tamaño es definido por width o height o según el contenido si éstos no estuvieran definidos

# 6. Propiedades flex items



- Propiedad flex
  - **flex**
    - Shorthand de las propiedades **flex-grow**, **flex-shrink** y **flex-basis** (en este orden)
    - Si se omiten algunos valores, los que toma flex son: **1** para flex-grow, **1** para flex-shrink y **0** para flex-basis
    - Ejemplos:
      - ✓ flex: 1 0 200px;
      - ✓ flex: 200px;
      - ✓ flex: 1;



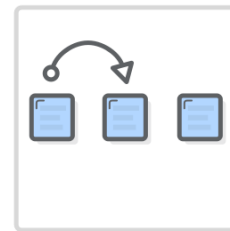
# 6. Propiedades flex items



- Orden de los elementos
  - **order**
    - Permite alterar el orden de los elementos flex
    - Por defecto todos los elementos tiene **order: 0**; (aparecen según su flujo HTML)
    - Primero van los valores inferiores
    - Es un cambio visual, ya que no afecta a la navegación por teclado y a los lectores de pantalla (usar con moderación)



**FLEX-DIRECTION**  
(WHOLE CONTAINER)

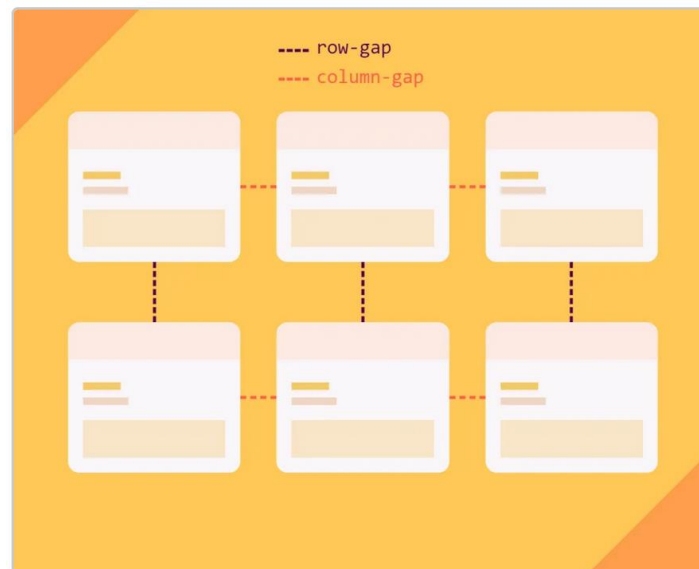


**ORDER**  
(INDIVIDUAL ITEMS)



# 7. Espaciado

- Gap
  - Shorthand de **row-gap** y **column-gap**
  - Define el espacio entre filas y columnas
  - Opción mejor que usar margin
  - Aplicable a Flexbox y CSS Grid



# 8. Anidación de contenedores



- Anidación de contenedores flex
  - Es posible anidar (agrupar) contenedores flex
  - Un elemento flex puede ser a su vez un contenedor flex



**NO GROUPING**

[3 FLEX ITEMS]



**GROUPED ITEMS**

[2 FLEX ITEMS]

# \*. Referencias



- Bibliografía y referencias
  - Libro “Diseño de Interfaces Web” de Eugenia Pérez Martínez / Pello Xabier Altadill Izura – Ed. Garceta
  - Curso Domestika “Layout web con CSS Grid y Flexbox y otras técnicas modernas”, de Javier Usobiaga Ferrer
  - Curso Openwebinars “Flexbox y CSS Grid” de Juan Diego Pérez
  - MDN web docs: <https://developer.mozilla.org/es/>
  - <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
  - <https://www.internetingishard.com/html-and-css/flexbox/>