MAQUETACIÓN WEB - FLEXBOX

Flexbox es un módulo completo de layout disponible en la especificación de CSS3. Define cómo se muestran los elementos y cómo se relacionan con el resto.
Podemos entender Flexbox como un modelo para la creación de layouts, que pretende mejorar los anteriores, aunque sin ser excluyente.

Flexbox es una herramienta muy avanzada para poder crear layouts de características avanzadas y necesarias en el día de hoy, donde es tan importante una estética cuidada y una gran adaptabilidad a distintos formatos de pantalla.

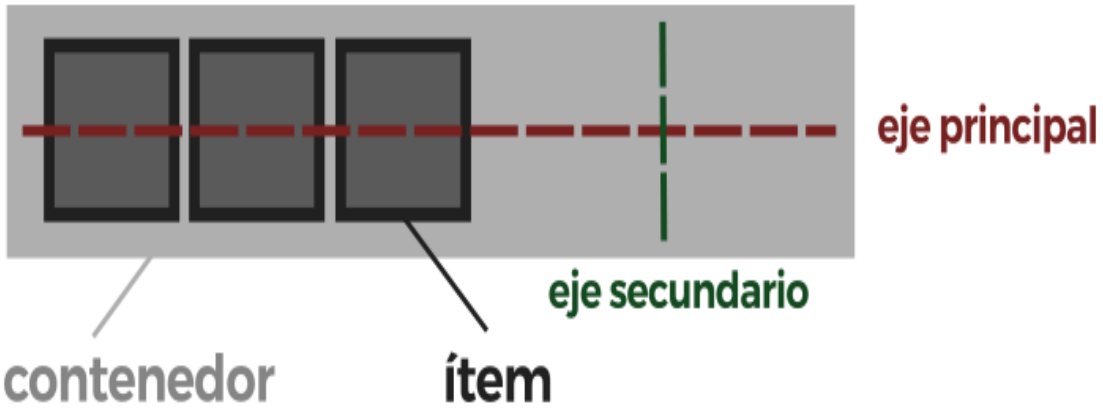
Flexbox agrega un nuevo tipo de "**display CSS**", con una completa gama de nuevas propiedades aplicables a ese tipo de display, a partir de los que puedes conseguir cosas extraordinarias. Por aclararnos, igual que tienes en CSS el display "block", "inline", "inline-block", etc. y sabías todos las propiedades que te acepta ese tipo de elementos, ahora dispones de "flex" e "inline-flex", siendo que los elementos que tienen ese nuevo display aceptan una cantidad enorme de nuevas propiedades de gran utilidad.

1. **FLEXBOX**

   Flexbox es un sistema de elementos flexibles, en la que los elementos HTML se adaptan y colocan automáticamente con lo cual va a ser más fácil personalizar los diseños. Está especialmente diseñado para crear, mediante CSS, estructuras de una sóla dimensión.

   En Flexbox diferenciamos dos elementos principales: **la caja contenedora** y **los elementos que situamos dentro**. Al aplicar un **display flex** o **display inline-flex** hacemos que una caja se comporte mediante este nuevo estándar y eso produce que los elementos que tiene como contenido se puedan distribuir con las propiedades de este estándar de maquetación.

   El contenedor va a poder modificar las dimensiones y el orden de los items, para acomodarlos de distintas maneras controladas por el desarrollador. Podremos repartir el espacio entre ellos de diversas formas, para distribuirlo a nuestro antojo, permitir que los items se estiren para ocupar todo el contenedor, o se encojan para que quepan en él sin desbordar, de colocarse distribuidos en filas o en columnas, etc.



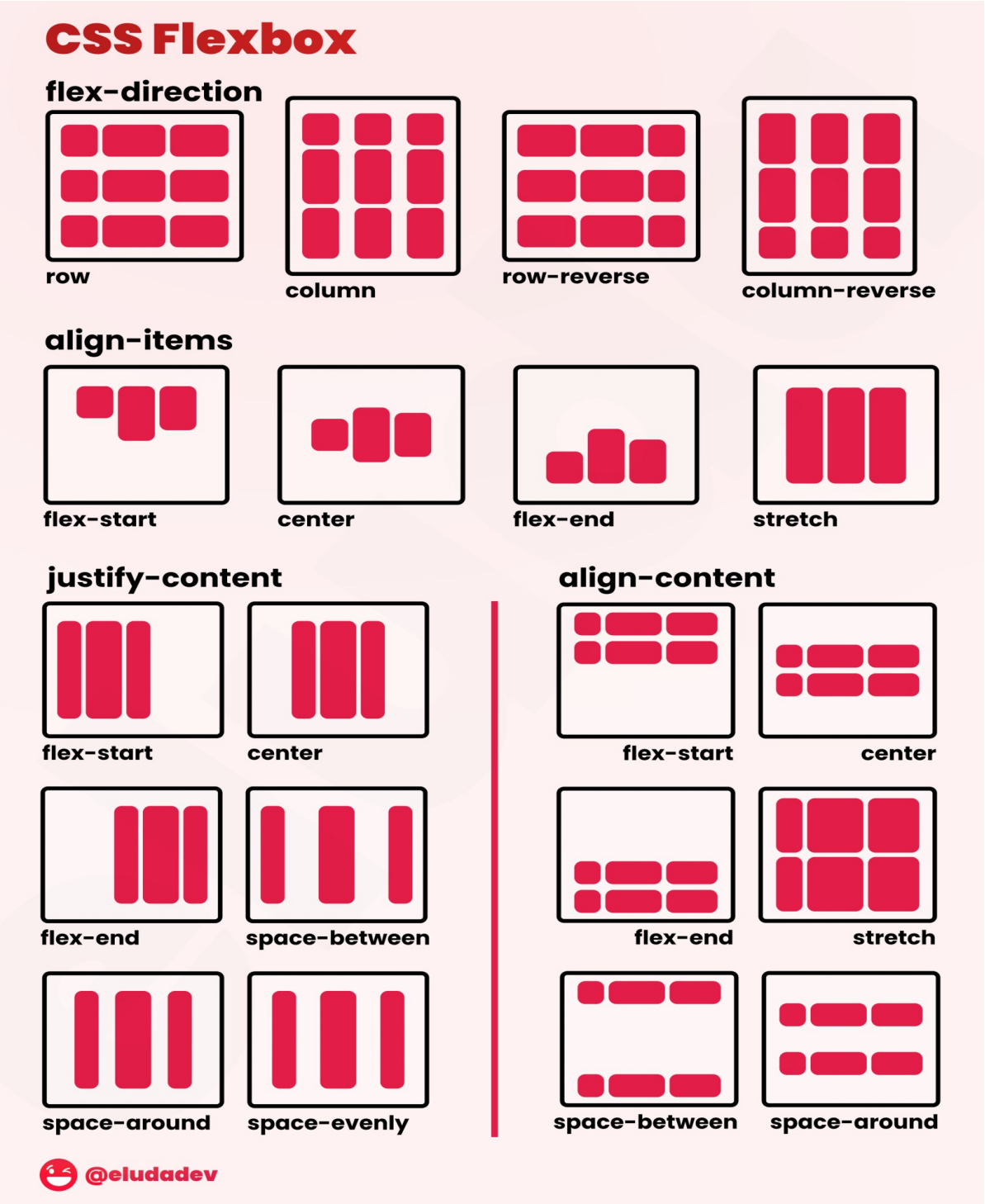
- **Contenedor:** Existe un elemento padre que es el contenedor que tendrá en su interior cada uno de los ítems flexibles y adaptables.
  - **Eje principal**: Los contenedores flexibles tendrán una orientación principal específica. Por defecto, es en horizontal (fila).
  - **Eje secundario:** De la misma forma, los contenedores flexibles tendrán una orientación secundaria, perpendicular a la principal. Si la principal es en horizontal, la secundaria será en vertical, y viceversa.
  - **Ítem**: Cada uno de los hijos flexibles que tendrá el contenedor en su interior.

Para activar el modo **flexbox** hay que utilizar sobre el elemento contenedor la propiedad **display** y especificar el valor **flex** o **inline-flex** dependiendo de como queramos que se comporte el contenedor: si como un elemento en línea, o como un elemento en bloque.

Tipo de elemento	Descripcion
inline-flex	Establece un contenedor de ítems flexible en línea, de forma equivalente a inline-block.
flex	Establece un contenedor de ítems flexible en bloque, de forma equivalente a block.

Por defecto, y sólo con esto, observaremos que los elementos se disponen todos sobre una misma línea. Esto ocurre porque estamos utilizando el modo **flexbox** y así estaremos trabajando con ítems flexibles básicos, garantizando que no se desbordarán

ni mostrarán los problemas que, por ejemplo, tienen los porcentajes sobre elementos que no utilizan flexbox.



Dirección de los ejes

Existen dos propiedades principales para manipular la dirección y comportamiento de los ítems a lo largo del eje principal del contenedor. Son las siguientes:

Propiedad	Valor	Significado
flex-direction:	**row**| row-reverse | column | column-reverse	Cambia la orientación del eje principal.
flex-wrap:	**nowrap**| wrap | wrap-reverse	Evita o permite el desbordamiento (multilinea).

Mediante la propiedad **flex-direction** podemos modificar la dirección del eje principal del contenedor para que se oriente en horizontal (por defecto) o en vertical. Además, también podemos incluir el sufijo **reverse** para indicar que coloque los ítems en orden inverso.

Valor	**Descripcion**
row	Establece la dirección del eje principal en horizontal.
row-reverse	Establece la dirección del eje principal en horizontal (invertido).
column	Establece la dirección del eje principal en vertical.
column-reverse	Establece la dirección del eje principal en vertical (invertido).

En general, flex se suele utilizar para estructuras de una sola dimensión, es decir, contenedores que sólo van en una dirección. Sin embargo, existe una propiedad denominada flex-wrap con la que podemos especificar un comportamiento especial del contenedor.

Por defecto, si un elemento no cabe dentro de nuestro contenedor flex, los elementos se harán más pequeños (son flexibles) para ajustarlos al contenedor. Este es el comportamiento por defecto de un contenedor flex. Sin embargo, con la propiedad **flex-wrap** podemos cambiar este comportamiento y permitir que nuestro contenedor flex se desborde, convirtiéndose en un contenedor **flex multilínea**

Valor	**Descripcion**
nowrap	Establece los ítems en una sola línea (no permite que se desborde el contenedor).

Valor	Descripcion
wrap	Establece los ítems en modo multilínea (permite que se desborde el contenedor).
wrap-reverse	Establece los ítems en modo multilínea, pero en dirección inversa.

Atajo: Dirección de los ejes

Recuerda que existe una propiedad de atajo (short-hand) llamada **flex-flow** con la que podemos resumir los valores de las propiedades **flex-direction y flew-wrap** especificándolas en una sola propiedad y ahorrándonos utilizar las propiedades concretas:

flex-flow: <flex-direction> <flex-wrap>

HUECOS (GAPS)

Existen dos propiedades de flexbox que han surgido recientemente: **row-gap** y **column-gap**. Dichas propiedades, permiten establecer el tamaño de un «hueco» entre ítems desde el elemento padre contenedor, y que eliminan la necesidad de estar utilizando padding o margin en los elementos hijos, con las complicaciones que ello suele conllevar:

Propiedad	Valor	Descripcion
row-gap	normal	Espacio entre filas (sólo funciona con flex-direction: column)
column-gap	normal	Espacio entre columnas (sólo funciona con flex-direction: row)

Flex es un sistema para diseños de una sola dimensión, por lo que sólo una de las dos propiedades tendrá efecto. Si la propiedad flex-direction está establecida en column, podrás utilizar row-gap, y en el caso de que la propiedad flex-direction se encuentre en row, podrás utilizar el column-gap.

Eso sí, es posible usar ambas si tenemos la propiedad flex-wrap definida a wrap y, por lo tanto, disponemos de multicolumnas flexbox, ya que en este caso si podemos separar elementos por filas y por columnas.

Los huecos sólo se aplican entre elementos, y no entre un elemento hijo y su contenedor padre.

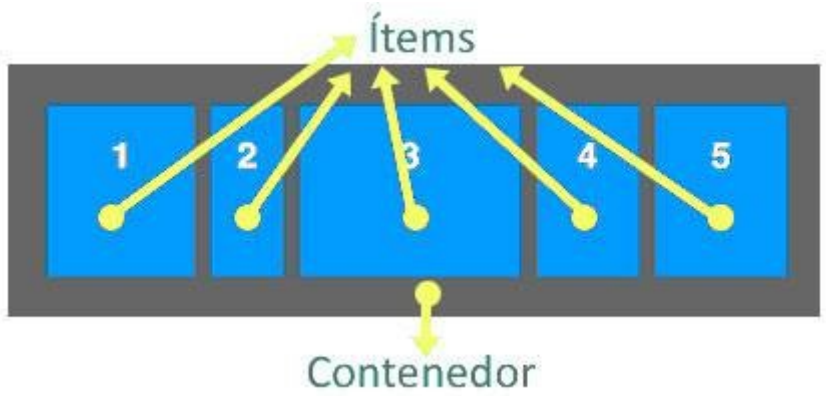
Atajo: Huecos

En Flex CSS existe una propiedad de atajo para los huecos, denominada **gap**. Con esta propiedad podemos indicar de una sola vez valores para las propiedades **row-gap** y **column-gap**, de forma que escribimos menos y es más cómodo en ciertas situaciones:

Propiedad	Valor	Descripcion
gap	0 | size	Aplica el tamaño indicado para el hueco en ambos ejes
gap	0 0 | size size	Aplica los tamaños indicados para el hueco del eje X y el eje Y

Como se puede ver, por defecto, el tamaño de los huecos es de 0, sin embargo, podemos utilizar tanto las propiedades individuales como la propiedad de atajo gap para modificar estos tamaños.

2. PROPIEDADES DE ALINEACIÓN DE ITEMS



Ahora que tenemos un control básico del contenedor de estos ítems flexibles, necesitamos conocer las propiedades existentes dentro de flexbox para disponer los ítems dependiendo de nuestro objetivo. Vamos a echar un vistazo a cuatro propiedades interesantes para ello:

Propiedad	Valor	Actúa sobre
justify-content:	**flex-start**| flex-end | center | space-between | space-around	Eje principal
align-content:	flex-start | flex-end | center | space-between | space-around | **stretch**	Eje principal

Propiedad	Valor	Actúa sobre
align-items:	flex-start | flex-end | center |**stretch**| baseline	Eje secundario
align-self:	**auto**| flex-start | flex-end | center | stretch | baseline	Eje secundario

Sobre el eje principal

La primera propiedad,**justify-content** sirve para colocar los ítems de un contenedor mediante una disposición concreta a lo largo del eje principal:

Valor	Descripcion
flex-start	Agrupa los ítems al principio del eje principal.
flex-end	Agrupa los ítems al final del eje principal.
center	Agrupa los ítems al centro del eje principal.
space-between	Distribuye los ítems dejando (el mismo) espacio entre ellos.
space-around	Distribuye los ítems dejando (el mismo) espacio a ambos lados de cada uno de ellos.

Una vez entendido este caso, debemos atender a la propiedad **align-content,** que es un caso particular del anterior. Nos servirá cuando estemos tratando con un contenedor flex multilinea, que es un contenedor en el que los ítems no caben en el ancho disponible, y por lo tanto, el eje principal se divide en múltiples líneas.

De esta forma **align-content** servirá para alinear cada una de las líneas del contenedor multilinea. Los valores que puede tomar son los siguientes:

Valor	Descripción
flex-start	Agrupa los ítems al principio del eje principal.
flex-end	Agrupa los ítems al final del eje principal.
center	Agrupa los ítems al centro del eje principal.
space-between	Distribuye los ítems desde el inicio hasta el final.
space-around	Distribuye los ítems dejando el mismo espacio a los lados de cada uno.

Valor	Descripción
stretch	Estira los ítems para ocupar de forma equitativa todo el espacio.

Sobre el eje secundario

La otra propiedad importante de este apartado es **align-items**, que se encarga de alinear los ítems en el eje secundario del contenedor. Hay que tener cuidado de no confundir **align-contents** con **align-items** , ***puesto que el primero actúa sobre cada una de las líneas de un contenedor multilinea (no tiene efecto sobre contenedores de una sola línea), mientras que align-items lo hace sobre la línea actual***. Los valores que puede tomar son los siguientes:

Valor	Descripción
flex-start	Alinea los ítems al principio del eje secundario.
flex-end	Alinea los ítems al final del eje secundario.
center	Alinea los ítems al centro del eje secundario.
stretch	Alinea los ítems estirándolos de modo que cubran desde el inicio hasta el final del contenedor.
baseline	Alinea los ítems en el contenedor según la base del contenido de los ítems del contenedor.

Por otro lado, la propiedad **align-self** actúa exactamente igual que **align-items**, sin embargo es la primera propiedad de flexbox que vemos que **se utiliza sobre un ítem hijo específico y no sobre el elemento contenedor.** Salvo por este detalle, funciona exactamente igual que **align-items**

Gracias a ese detalle,**align-self** nos permite cambiar el comportamiento de **align-items** y sobreescribirlo con comportamientos específicos para ítems concretos que no queremos que se comporten igual que el resto. La propiedad puede tomar los siguientes valores:

Valor	Descripción
flex-start	Alinea los ítems al principio del contenedor.
flex-end	Alinea los ítems al final del contenedor.
center	Alinea los ítems al centro del contenedor.
stretch	Alinea los ítems estirándolos al tamaño del contenedor.
baseline	Alinea los ítems en el contenedor según la base de los ítems.
auto	Hereda el valor de **align-items** del padre (o si no lo tiene, **stretch**).

Si se especifica el valor auto a la propiedad **align-self**, el navegador le asigna el valor de la propiedad **align-items** del contenedor padre, y en caso de no existir, el valor por defecto:stretch.

Propiedades de ítems hijos

A excepción de la propiedad **align-self** todas las propiedades que hemos visto hasta ahora se aplican sobre el elemento contenedor. Las siguientes propiedades, sin embargo, se aplican sobre los ítems hijos. Echemos un vistazo:

Propiedad	Valor	Descripción
flex-grow:	**0** |*[factor de crecimiento]*	Número que indica el crecimiento del ítem respecto al resto.
flex-shrink:	**1**|*[factor de decrecimiento]*	Número que indica el decrecimiento del ítem respecto al resto.
flex-basis:	**TAMAÑO**|**content**	Tamaño base de los ítems antes de aplicar variación.
order:	**0**|*[número]*	Número (peso) que indica el orden de aparición de los ítems.

Atajo: Propiedades de ítems hijos

Existe una propiedad llamadaa **flex** que sirve de atajo para estas tres propiedades de los ítems hijos. Funciona de la siguiente forma:

```
.item {
```

```
    /* flex: <flex-grow> <flex-shrink> <flex-basis> */
    flex: 1 3 35%;
  }
```

- **Propiedad Flex-grow**

Esta propiedad sirve para indicar cómo de grandes y cómo se van a distribuir los ítems dentro del contenedor. Con el siguiente ejemplo podemos ver cómo actúa esta propiedad:

Como vemos en la imagen, dependiendo de los valores indicados en los *flex-grow*, los ítems se muestran con diferentes anchuras.
Los ítems con el valor ***flex-grow: 2*** ocupan el doble que los que tienen **flex-grow: 1**. Mientras que el que tiene ***flex-grow: 3***es exactamente el triple.

- **Propiedad align-self:** Con esta propiedad, podemos modificar el valor de align-items que hayamos podido dar a los ítems en el contenedor principal y alinearlos a la vez, en cambio, con align-self lo haremos en uno concreto.

En esta imagen se ver más claro:

- **Propiedad *flex-shrink:*** Nos vale para indicar qué ítems encogerán su tamaño. Su valor por defecto es 1 y darle un valor más alto significa que el elemento se encogerá respecto a lo que ocuparía sin asignárselo.

- **Propiedad *flex-basis:*** Con esta propiedad modificamos las dimensiones de los elementos, pero antes de que el espacio que sobra sea distribuido, por razones que pueda haber con otras otras propiedades, como ***flex-grow***.

Estos son los valores que podemos otorgarle:

- **Número, unidad CSS o porcentaje:** las dimensiones iniciales del elemento serán indicadas con este parámetro y previo de conceder el espacio que sobra.
- **Auto: e**s lo que tiene por defecto, lo que significa que no tendrá ningún efecto y que deja el dimensionamiento a otro atributo que pudiera tener.

- **Orden de los ítems**

Por último, y quizás una de las propiedades más interesantes, es **order** que modifica y establece el orden de los ítems según una secuencia numérica.

Por defecto, todos los ítems flex tienen un **order:0** implícito, aunque no se especifique.

Si indicamos un **order** con un valor numérico, irá recolocando los ítems según su número, colocando antes los ítems con número más pequeño (incluso valores negativos) y después los ítems con números más altos.

De esta forma podemos recolocar fácilmente los ítems incluso utilizando media queries o responsive design.