FLEXBOX

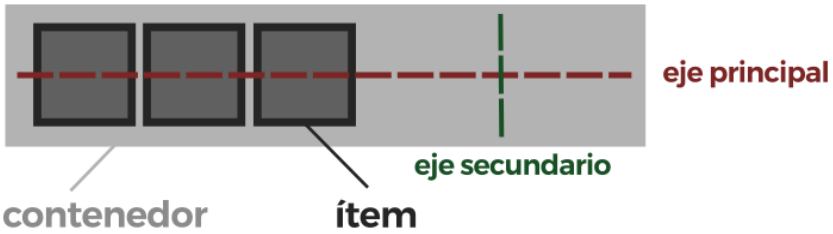
La propiedad **Flexible Box,** o **flexbox**, de CSS3 es un modo de diseño que permite colocar los elementos de una página para que se comporten de forma predecible cuando el diseño de la página debe acomodarse a diferentes tamaños de pantalla y diferentes dispositivos

Tradicionalmente, en CSS se ha utilizado el posicionamiento (static, relative, absolute...), los elementos en línea o en bloque (y derivados) o los float, lo que a grandes rasgos no dejaba de ser un sistema de creación de diseños bastante tradicional que no encaja con los retos que tenemos hoy en día (sistemas de escritorio, dispositivos móviles, múltiples resoluciones, etc...).

Flexbox es un sistema de elementos flexibles que llega con la idea de olvidar estos mecanismos y acostumbrarnos a una mecánica más potente, limpia y personalizable, en la que los elementos HTML se adaptan y colocan automáticamente y es más fácil personalizar los diseños.

Conceptos

Para empezar a utilizar flexbox lo primero que debemos hacer es conocer algunos de los elementos básicos de este nuevo esquema, que son los siguientes:

- 

Contenedor: Existe un elemento padre que es el contenedor que tendrá en su interior cada uno de los ítems flexibles y adaptables.

- **Ítem**: Cada uno de los hijos flexibles que tendrá el contenedor en su interior.
- **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.

Contenedor

El elemento "padre" que contiene los elementos flexibles. Un contenedor flexible se define usando los valores flex o inline-flex en la propiedad **display**.

Elemento flexible (Flex item)

Cada hijo de un contenedor flex se convierte en un elemento flexible. Si hay texto directamente incluido en el contenedor flexible, se envuelve automáticamente en un elemento flexible anónimo.

Ejes

Cada diseño de "caja flexible" sigue dos ejes. El **eje principal** es el eje a lo largo del cual los elementos flexibles se suceden unos a otros. El **eje secundario** es el eje perpendicular al **eje principal**.

- La propiedad flex-direction establece el eje principal.
- La propiedad justify-content define cómo los elementos flexibles se disponen a lo largo del eje principal en la línea en curso.
- La propiedad align-items define cómo los elementos flexibles se disponen a lo largo del eje secundario de la línea en curso.
- La propiedad align-self define cómo cada elemento flexible se alinea respecto al eje secundario, y sustituye al valor por defecto establecido por align-items.

Los lados **inicio principal/fin principal (main start/main end)** e **inicio secundario/fin secundario (cross start/cross end)** del contenedor flexible describen el origen y final del flujo de los elementos flexibles.

- La propiedad `order` asigna elementos a grupos ordinales y determina qué elementos aparecen primero.

Unas web muy buenas para poder visualizar estas explicaciones son:

- https://demo.agektmr.com/flexbox/
- http://the-echoplex.net/flexyboxes
- http://flexbox.help/
- http://flexboxfroggy.com/

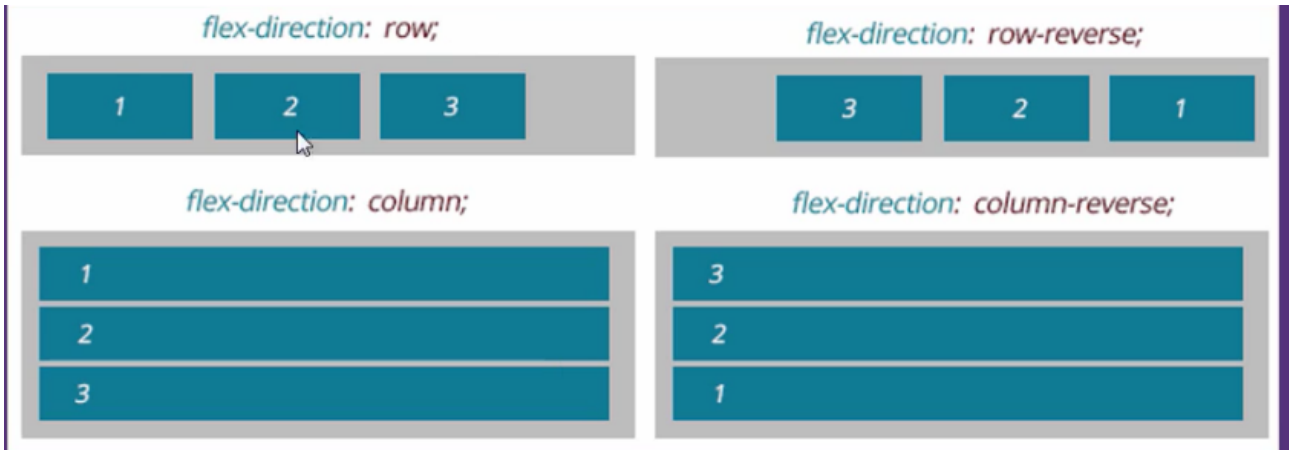
SIGUE LAS SIGUIENTES REGLAS PARA UN DISEÑO FLEXBOX

REGLA 1:
FLEXBOX NECESITA AL MENOS UN CONTENEDOR PADRE Y UN CONTENEDOR HIJO

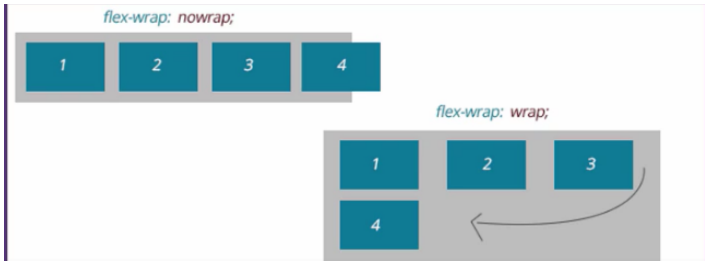
Con la propiedad display:flex del padre: los hijos tendrán la altura del padre y la anchura dependerá del contenido de cada hijo.

REGLA 2:
EL CONTENEDOR FLEXIBLE(FLEX CONTAINER) TIENE DOS EJES.EL EJE PRINCIPAL QUE ES EL EJE HORIZONTAL , Y EL EJE VERTICAL

LA ETIQUETA flex-direction:row o column, determina la orientación del eje principal. Por defecto es row



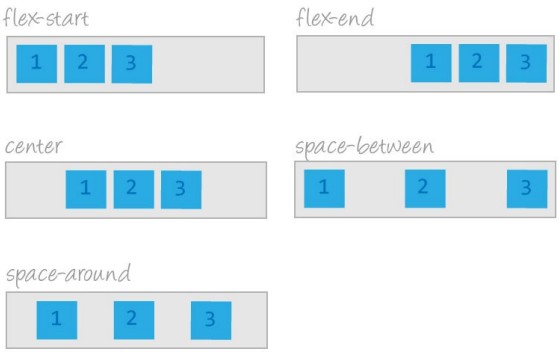
REGLA 3.
SE PUEDE PERMITIR EL SALTO DE FILA CON LA PROPIEDAD
flex-wrap:nowrap o wrap



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

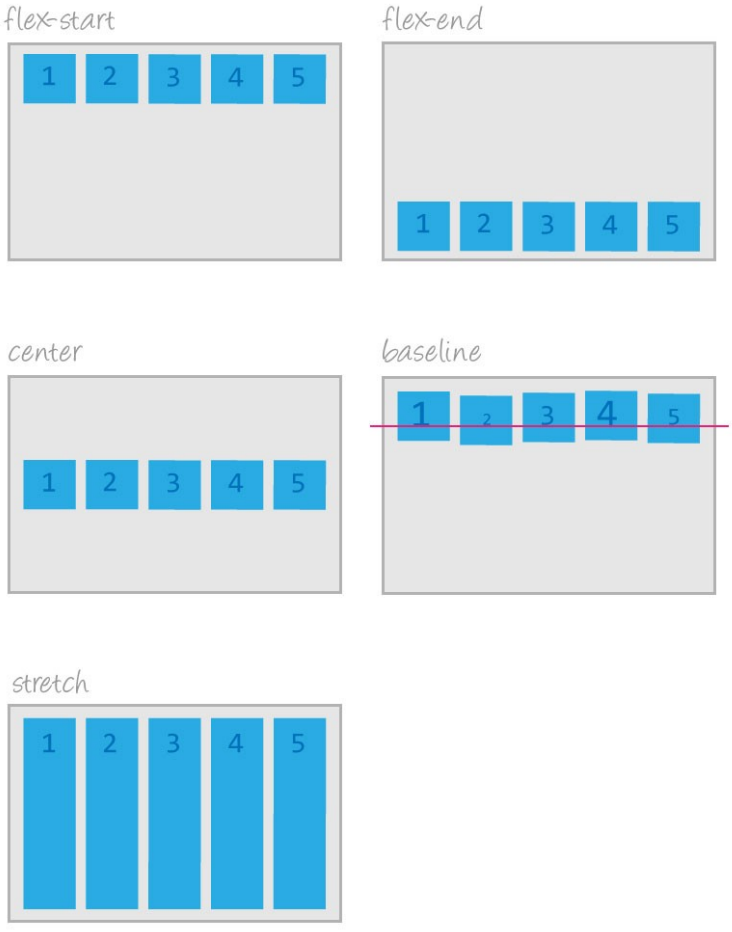
```
flex-wrap: <flex-direction> <flex-wrap>; */
flex-wrap: row wrap;
```

REGLA 4.
ALINEAMOS ELEMENTOS EN EL EJE PRINCIPAL CON LA PROPIEDAD
justify-content:



Justify-Content

REGLA 5.
ALINEAMOS ELEMENTOS EN EL EJE SECUNDARIO CON LA PROPIEDAD
align-items



Align-Items

REGLA 6.
ALINEAMOS ELEMENTOS HIJOS DE FORMA INDIVIDUAL EN EL EJE SECUNDARIO CON LA PROPIEDAD.
Align-seLf

En este caso se le puede asignar una alineación concreta a un elemento hijo y sobreescribe la propiedad indicada en align-items.

REGLA 7.
LOS HIJOS FLEXIBLES IGNORAN PROPIEDADES COMO float, clear,vertical-align

Pero no hay que olvidar que el contenedor padre se comporta como un contenedor cualquiera, es decir, para éste sí se puede usar cualquiera de estas propiedades.

REGLA 8.
SE PUEDE MODIFICAR EL TAMAÑO DE LOS CONTENEDORES HIJOS CON LAS PROPIEDADES:

- flex-grow
- flex-shrink
- flex-basis

No se puede olvidar que, por defecto, el tamaño del contenedor hijo depende de su contenido

- flex-grow--→define el tamaño que crecerá, del espacio disponible, un hijo en relación a los demás hermanos.
  Por defecto el valor es 0
  Si todo tienen el valor 0 no se reparten nada pero si uno de los hermanos tiene el valor a 1 y el resto a 0 , éste se quedará con todo el espacio libre.
  Si dos de ellos tienen el valor 1, se repartirán el espacio libre entre esos dos y así sucesivamente.
  Si alguno de los hermanos tiene este valor superior significará que la proporción del espacio sobrante será el doble , el triple…. Que el que tiene el valor 1.
- flex-shrink-→ define el tamaño que encoje dicho contenedor.
  Si el espacio disponible es negativo (el tamaño del contenedor es menor a la suma de los tamaños de los items), de forma predeterminada los items se encogen en proporciones iguales para caber en una sola línea (en un próximo artículo trabajaremos flexbox con varias líneas de items).
- Flex-basis-→
  La propiedad `flex-basis`, como su nombre lo indica, define el tamaño base principal para un flex-item. Esto significa que no necesariamente ese será su tamaño al dibujarse por el navegador, pero que será un punto de partida para calcular el tamaño final.

  En otras palabras, si el main axis es horizontal (predeterminado), flex-basis será equivalente a width; y si el main axis es vertical, flex-basis será equivalente a height.

  Ten en cuenta que el tamaño definido por flex-basis es, como su nombre lo dice, el tamaño base. Es decir, que podrá variar (crecer o encogerse), según los valores de flex-grow y flex-shrink que veremos más adelante.

Importante

•flex-basis siempre gana sobre el valor de width o height.

•Si no se define el valor de flex-basis o se establece en auto, se tomará en cuenta el valor de width o height según sea el caso.

•Si no se define un valor para flex-basis y tampoco se especifica el tamaño por width o height, se definirá el main-size según su contenido.

-

Existe una propiedad llamada 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%
}
```