FLEXBOX

**Explicar el starter**

Tenemos container con anchura de 90vw y no tiene altura. Así que su altura es la forzada por su contenido. Contiene 10 **divs** (box) que tampoco sin anchura así que son del 100% de su contenedor (container) y sin altura (así que forzada por su contenido)

**Explicar Flexbox y sus conceptos**

Flexbox es un módulo de CSS que fue recomendación candidata en noviembre de 2018

https://www.w3.org/TR/css-flexbox/

((Nota: El diseño de caja flexible es más apropiado para los componentes de una aplicación y diseños de pequeña escala, mientras que ***GRID*** está diseñado para diseños de mayor escala, como la estructura general de la página.))

Tiene como objetivo proporcionar una forma más eficiente de diseñar, alinear y distribuir el espacio entre los elementos en un contenedor, incluso cuando su tamaño es desconocido y / o dinámico.

La combinación de **display** (block, inline, inline-block) y **position** se queda corta para abordar los retos de diseño actuales.

A diferencia de **block** que tiene claramente una dirección vertical, de **inline** claramente una dirección u orientación horizontal, **Flexbox** es independiente de la dirección, funciona bien en ambas.

**FlexBox** no es una simple propiedad, es un conjunto de propiedades relacionadas a las que vamos a llamar modulo.

Algunas definiciones...

*Contenedor*: Es el elemento padre que tendrá en su interior cada uno de los ítems flexibles. Observa que al contrario que muchas otras estructuras CSS, por norma general, en Flex establecemos las propiedades al elemento padre.

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

Eje principal: Los contenedores flexibles tendrán una orientación principal específica. Es la direccion en la que se disponen sus flex-items. Por defecto, es en horizontal (en 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.

**1. Para activar el modo flexbox en un contenedor usamos la propiedad display: flex;**

**.container {**

**display: flex | inline-flex**

**}**

**inline-flex** Establece un contenedor en línea, similar a inline-block (ocupa solo el contenido).

**flex** Establece un contenedor en bloque, similar a block (ocupa todo el ancho del padre).

Al hacer a container un contenedor flex activamos aun sin mencionarlas algunas propiedades por defecto.

Los flex-ítems se disponen en horizontal.

La altura del contenedor se reduce, pero porque sigue siendo la altura de su contenido

Si reducimos la ventana del navegador se observa que los flex-ítems pueden desbordar el contenedor

Si estableciésemos la altura+ de los flex-ítems, container crecería

\* Si estableciésemos la altura+ de container, los flex-ítems también crecerían porque por defecto los flex-ítems ocupan la altura del contenedor

\* Flexbox puede reducir la anchura de los flex-items para acomodarlos al contenedor (pero nunca mas de lo que permita el contenido de estos)  
Por ejemplo si ponemos una anchura de 200px (a los flex-items) estos llenaran el contenedor, (se ampliaran hasta intentar alcanzar los 200). Una vez alcanzados los 200 no crecen (jugar con la ventana del navegador)

Por ejemplo si ponemos una anchura del 50%, cada flex-item debria ocupar la mitad, pero como aun no hemos establecido flex-wrap, simplemente llenaran el contenedor. Al poner wrap, cada uno ocupara el 50% y se dispondrán en varias líneas.

== Por defecto los flex-items intentan caber todos en una linea

**2. flex-direction**

Cambia la orientación del eje principal.

**.container {**

**flex-direction: row | row-reverse | column | column-reverse;**

**}**

**3. flex-wrap**

Por defecto los flex-items intentan caber (mantenerse) todos en una sola linea

Sin embargo, si especificamos wrap en la propiedad flex-wrap, lo que permitimos es que el contenedor se pueda desbordar, pasando a ser un contenedor multilínea

**.container {**

**flex-wrap: nowrap | wrap | wrap-reverse;**

**}**

**4. flex-flow**

Es un propiedad de atajo para especicificar flex-direction y flex-wrap a la vez.

**.container {**

**/\* flex-flow: <flex-direction> <flex-wrap>; \*/**

**flex-flow: row wrap;**

**}**

**5.**