



# Universidad Don Bosco

---

## Lenguajes Interpretados en el Cliente G02T

### Tarea ex aula

#### **Docente:**

Guillermo de Jesús Calderón Hernández

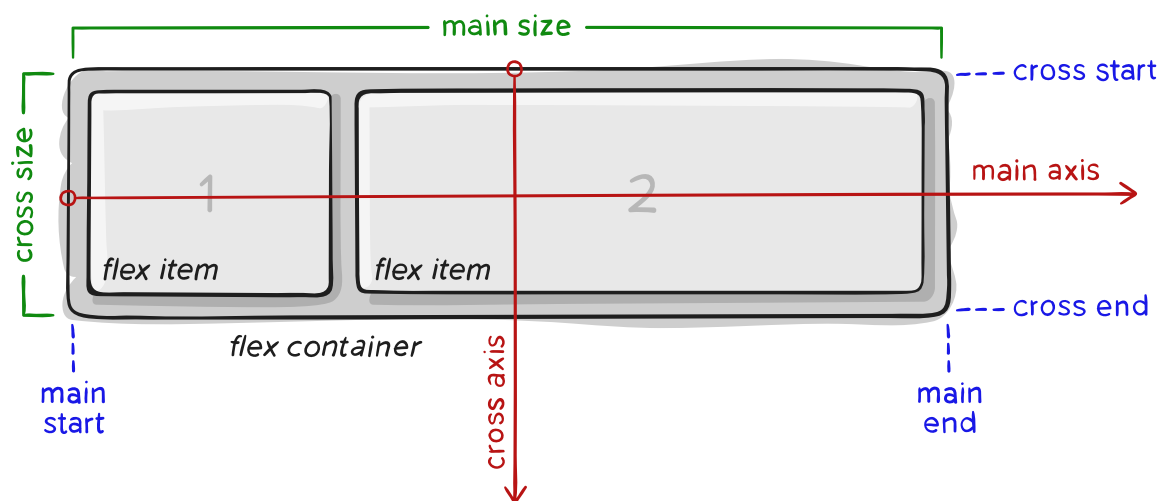
#### **Integrantes:**

- Justin Alfredo Vásquez Alfaro VA211514
- Carlos Alejandro Reyes Segovia RS190102
- Angel Guillermo Sanchez Mangandi SM192656

Soyapango, sábado 09 de septiembre, 2023

## Flexbox

El módulo de caja flexible es una nueva técnica de maquetación con css, que me permite crear un layout y organizar los elementos de mejor manera en una página, diseñado como un modelo unidimensional de diseño y como un método que pueda ayudar a distribuir el espacio entre los elementos de una interfaz y mejorar las capacidades de alineación. Antes del Flexbox las únicas herramientas disponibles para crear diseños css y posicionar elementos con buena compatibilidad entre navegadores eran flotar y posición, algunas tareas que consideramos básicas al momento de crear una página como centrar verticalmente un elemento secundario en relación con un elemento principal o hacer que los elementos ocupen la misma cantidad de espacio, o que las columnas tengan el mismo tamaño independiente de la cantidad de contenido interno, resultan bastante tediosas con estas 2 herramientas aplicarlas de forma práctica o “flexible”, pues la herramienta Flexbox fue creada para hacer estas tareas más sencillas y funcionales. Flexbox es un módulo completo y no una propiedad única; algunos de ellos deben declararse en el contenedor llamado Flex container (contenedor flexible) mientras que otros deben declararse en los elementos secundarios llamados Flex items. El siguiente diagrama ejemplifica la idea central del diseño Flex:



Los elementos principales de Flexbox son:

- Main axis: el eje principal de un contenedor flexible es el eje principal a lo largo del cual se colocan los elementos flexibles, este define el ancho del Flex ítem. Estos contenedores se insertan iniciando por el lado start (izquierda) hasta el lado end (final), tanto el Main axis como el Cross axis se pueden modificar su propiedad de a lo largo por horizontal o verticalmente con su propiedad flex-direction.
- Cross axis: es el eje perpendicular al eje principal, este define las columnas y el largo de los elementos flexibles dentro del Flex container.

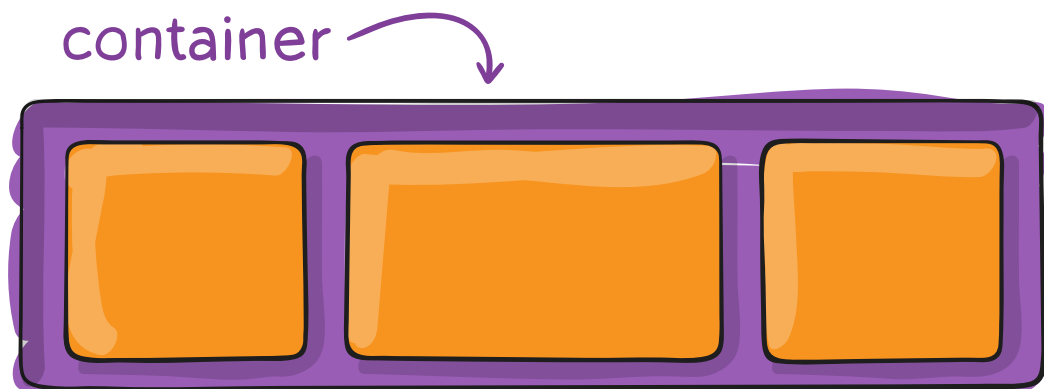
## Flex Container

El contenedor flexible es el área del documento que contendrá los Flexbox, al establecer la propiedad display en ellos, los ítems dentro de ellos se convertirán automáticamente en Flex ítems, estos se comportan de la siguiente manera:

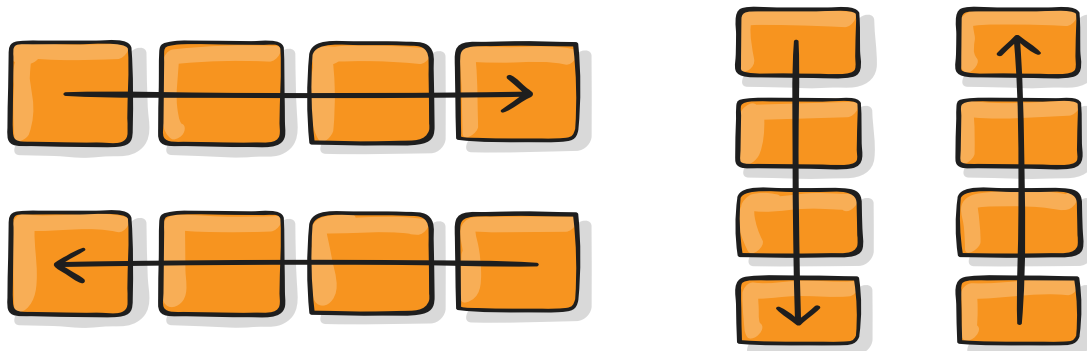
- los ítems se despliegan sobre una fila.
- estos inician desde el margen inicia sobre el eje principal.
- los ítems se ajustan para obtener el tamaño del eje cruzado.

Por enumerar algunas de las principales propiedades de estos contenedores flexibles tenemos:

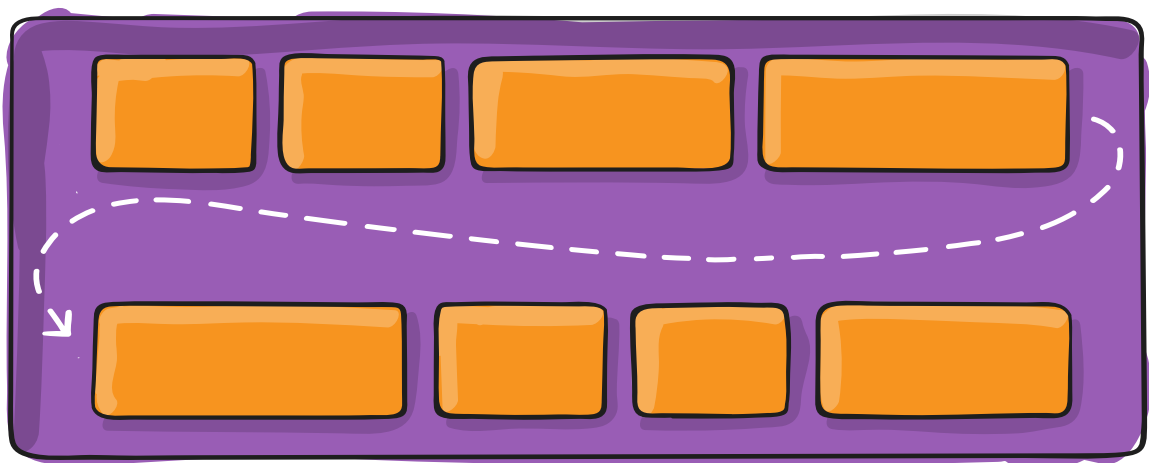
- Display: esta propiedad define un Flex container; coloca todos los elementos secundarios directos en un contexto Flex, el resultado es que todos los ítems se alinearán en una solo fila, usando el tamaño del contenedor como su tamaño en el eje principal. Si hay más ítems de los que caben en el contenedor, estos no pasarán más abajo si no que sobrepasarán el margen. Si hay ítems más altos que otros, todos los ítems serán ajustados en el eje cruzado para alcanzar al mayor. Un vistazo rápido de esta propiedad sería el siguiente:



- Flex-direction: esta propiedad establece el eje principal definiendo así la dirección de los elementos que se colocan dentro del contenedor flexible, se considera que la Flexbox es un concepto de diseño unidireccional (un vector), y los elementos dentro de la misma estuvieran dispuestos principalmente en filas o columnas. Al aplicar esta propiedad al contenedor flexible, puede ocurrir que el eje principal pase de definir las filas a columnas, o que las líneas de inicio y fin de cada eje respectivamente fueran puestas al revés, tal y como se especifica en la siguiente figura:

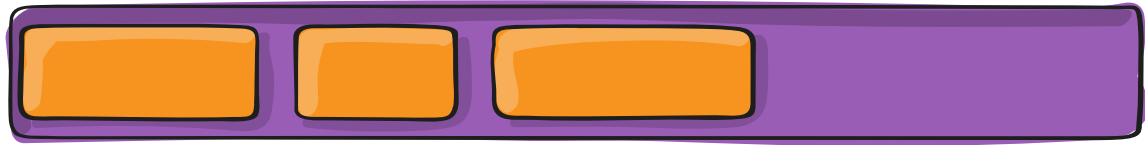


- Flex-wrap: En general Flex establece que se utiliza para estructuras de una sola dimensión, es decir, contenedores que van en una sola dirección, sin embargo, existe una propiedad denominada Flex-wrap con la que se puede especificar un comportamiento especial dentro del contenedor. Esta propiedad cambia el comportamiento de los Flex ítems de ajustarse todos en una sola línea a ordenarse en 2 o más líneas según sea necesario. También esta su contraparte “nowrap”, que no permite que los Flex ítems pasen a la siguiente línea, el cual viene por defecto en las propiedades de la Flexbox.



- Justify-content: esta propiedad define la alineación de los ítems a lo largo del eje principal. Ayuda a distribuir el espacio libre que queda dentro del contenedor, también ejerce cierto control sobre la alineación de los ítems cuando sobrepasan el límite de la línea, teniendo en cuenta la compatibilidad del navegador con estos valores tiene matices por lo que puede variar el tamaño del espaciado o la distancia entre los bordes y el contenido. Siendo los más destacados:
  - Flex-start: los ítems se alinean a lo largo del borde de inicio, según sea la dirección del contenedor.
  - Flex-end: los ítems se alinean a lo largo del borde final, según sea la dirección del contenedor.
  - center: los ítems están centrados en la línea.

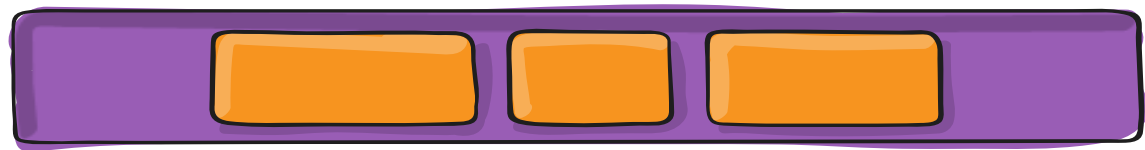
flex-start



flex-end

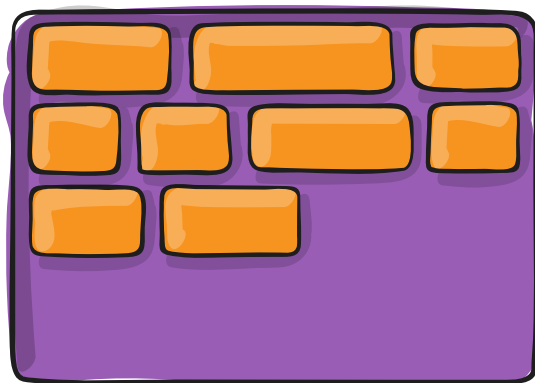


center

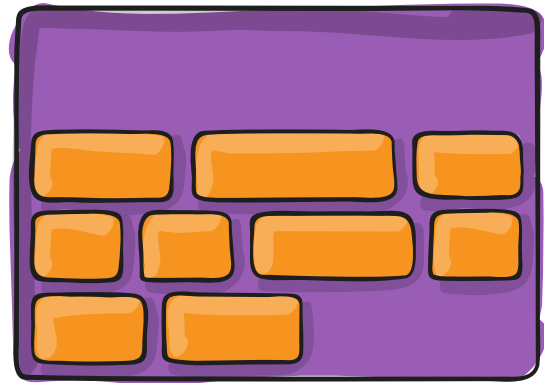


- Align-items: Al aplicar esta propiedad al Flex container la misma alineará a los elementos con respecto al eje cruzado. Inicialmente su valor estará dado como estándar, razón por la cual los ítems se ajustan por defecto a la altura del elemento más alto, esto para llenar el contenedor flexible, es decir, el ítem más alto define la altura de este. En cierto modo funciona de manera similar al Justify-content, pero en vez del eje principal toma como referencia el eje transversal.

flex-start



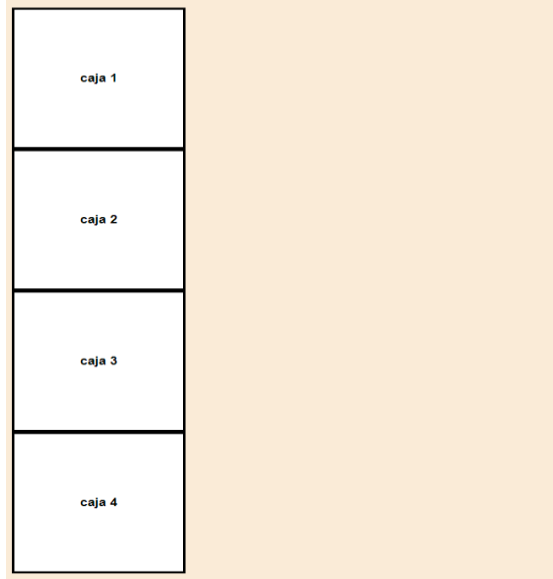
flex-end



De manera práctica se presentan algunos ejemplos de cómo aplicar estas propiedades.

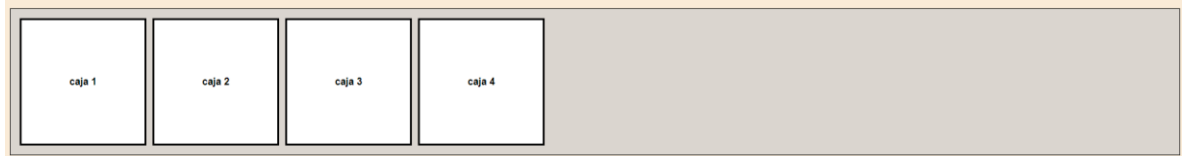
Para este caso se han declarado 4 DIVS dentro de un contenedor con sus respectivas propiedades; 200px, letra negrita, arial, texto centrado y borde negro.

#### aplicacion de las propiedades para las Flexbox



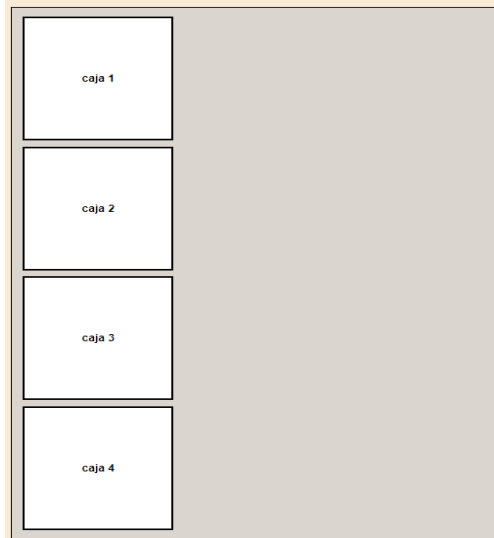
Al aplicar el comando `display: Flex`, se alinearán de forma horizontal de manera automática, también se destaca el Flexbox.

#### aplicacion de las propiedades para las Flexbox

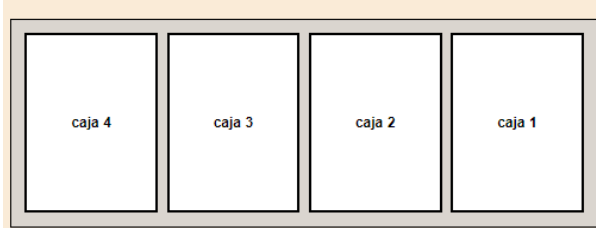


El Flexbox tiene por defecto que sus elementos se organicen en filas, para cambiar eso, se puede aplicar la propiedad `flex-direction: column` o bien `flex-direction: row-reverse`.

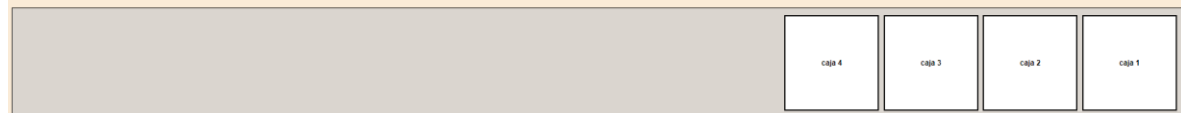
#### aplicacion de las propiedades para las Flexbox



#### aplicacion de las propiedades para las Flexbox



#### aplicacion de las propiedades para las Flexbox



Para que las cajas ocupen el 100% del espacio o alinearlas según me convenga, se utiliza la propiedad `Justify-content`.

aplicacion de las porpiedades para las Flexbox

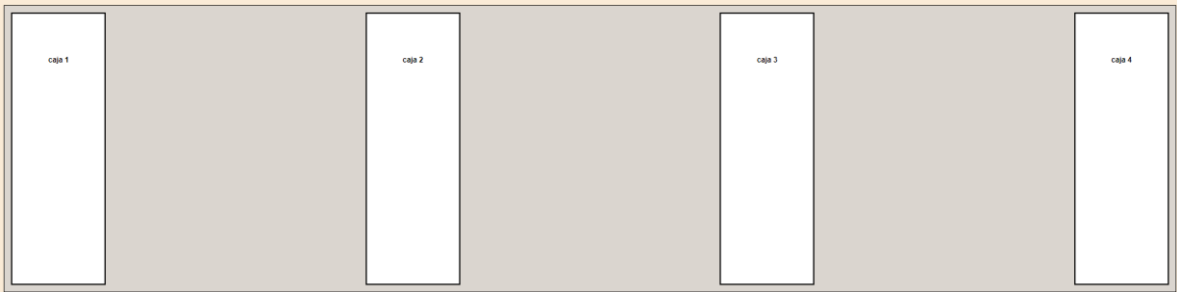


aplicacion de las porpiedades para las Flexbox



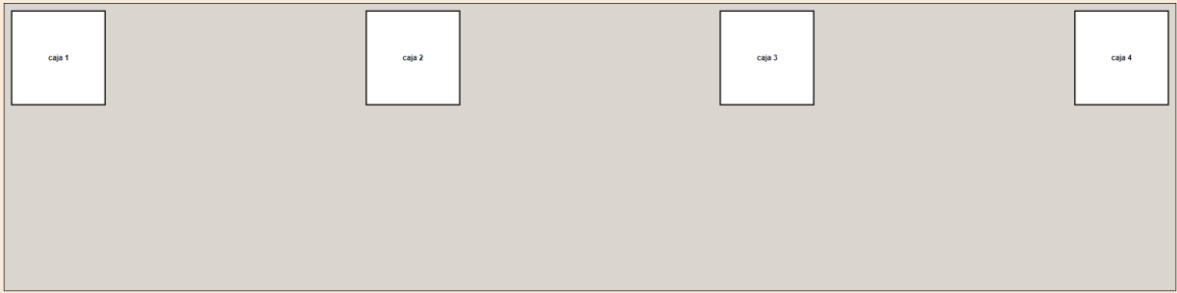
En el caso de que el contenedor sea mucho más grande que las cajas, estas últimas se alinearan según el tamaño de esta.

aplicacion de las porpiedades para las Flexbox



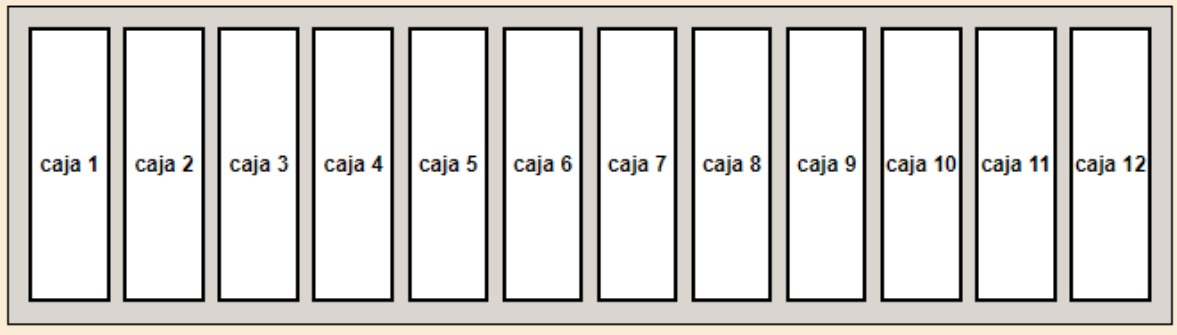
Para organizar esto en el caso de que se quiera alinear hacia arriba, se aplica la propiedad `Align-items: Flex-start`

aplicacion de las porpiedades para las Flexbox

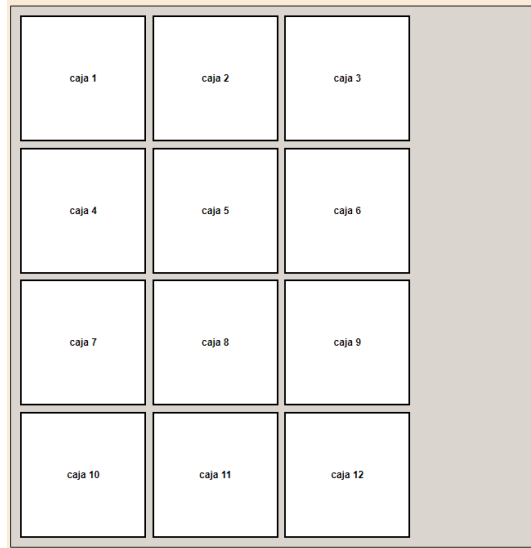


Al agregar más cajas, estas se irán acomodando al tamaño del Flexbox, para evitar eso se pueden agrupar en más líneas con el comando `Flex-wrap: wrap`.

## aplicacion de las porpiedades para las Flexbox



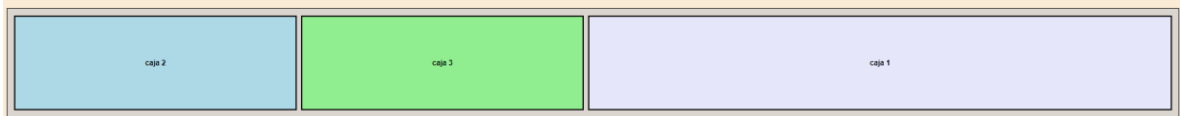
#### aplicacion de las propiedades para las Flexbox



Para terminar con las propiedades, a cada caja se le puede asignar al tamaño y el orden que pueden ocupar dentro de la Flexbox con los comandos flex-grow y order respectivamente, a la vez se pueden combinar con la propiedad Flex-basis que aplica lo mismo, pero en porcentajes, para este ejemplo se usan las primeras 3 cajas como se muestra en la imagen.

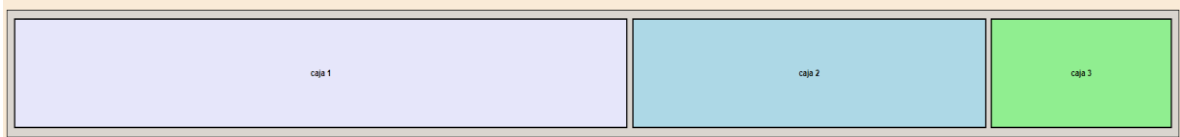
Caja 1 ocupa el 50% del Flex

#### aplicacion de las propiedades para las Flexbox



Caja 1 tiene prioridad sobre las otras 2, así mismo la caja 2 tiene prioridad sobre la 3

#### aplicacion de las propiedades para las Flexbox



## Grid css Layout

Es un sistema de diseño bidimensional basado en cuadrículas que, en comparación con cualquier sistema de diseño de páginas web del pasado, cambia por completo la forma en la que diseñamos las interfaces del usuario, css siempre se ha utilizado para diseñar nuestras páginas web, pero nunca ha hecho un buen trabajo, primero se usan tablas, luego flotantes, posicionamiento y bloques en línea, pero todos estos métodos fueron esencialmente trucos y omitieron muchas funciones importantes. Flexbox es una excelente herramienta de diseño, pero su flujo unidireccional tiene diferentes casos de uso y, de hecho, funcionan juntos. Grid es el primer módulo css creado específicamente para resolver problemas de diseño que todos hemos estado solucionando desde que se crean los sitios web. Este sistema de cuadrículas se destaca por permitir una página en

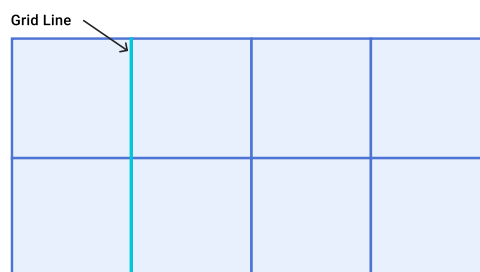


áreas o regiones principales, por definir la relación en términos de tamaño, posición y capas entre partes de un control construido a partir de primitivas HTML.

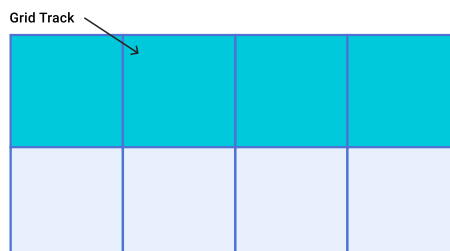
Al igual que las tablas, el Grid layout permite al autor alinear elementos en filas y columnas como matrices, sin embargo, con css Grid son posibles muchos más diseños y de forma más sencilla que con las tablas, dado que es posible asignarle tamaño, posición o colocación, colocando el nombre de sus clases creando así cuadrículas flexibles y potentes de forma prácticamente instantánea con la nueva familia de propiedades CSS. De manera similar a Flexbox, el orden de origen de los elementos de la cuadrícula no importa. Su CSS puede colocarlos en cualquier orden, lo que hace que sea muy fácil reorganizar su cuadrícula con consultas de medios. Imagine definir el diseño de toda su página y luego reorganizarla completamente para acomodar un ancho de pantalla diferente, todo con solo un par de líneas de CSS.

Para comenzar se debe definir un elemento contenedor como una cuadrícula con el comando `display: Grid`, establecer los formatos de columna y fila con `grid-template-columns` y `grid-template-rows`, y luego colocar sus elementos en la cuadrícula con `Grid-column` y `Grid-row`. Para entender el método y la práctica de css Grid, se debe repasar primero su terminología:

- **Línea de cuadrículas:** Una cuadrícula está formada por líneas, que corren horizontal y verticalmente. Si su cuadrícula tiene cuatro columnas, tendrá cinco líneas de columna, incluida la que está después de la última columna. Las líneas se numeran empezando por 1, y la numeración sigue el modo de escritura y la dirección del script del componente.

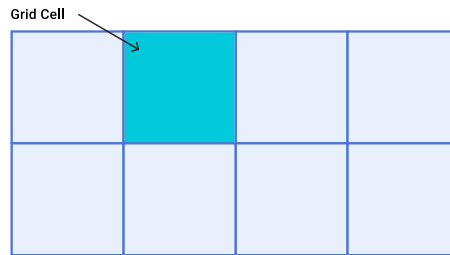


- **Bandas de cuadrícula:** una banda es el espacio entre 2 líneas de cuadrícula. Una banda de fila está entre 2 líneas de fila y una banda de columna. Cuando creamos nuestra cuadrícula, creamos estas bandas asignándoles un tamaño.

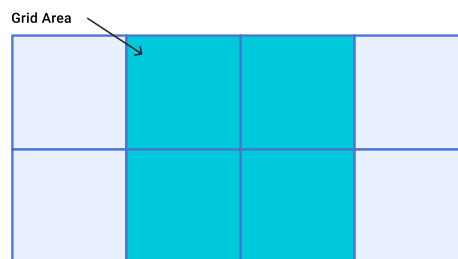


- **Celda de cuadrícula:** Una celda de cuadrícula es el espacio más pequeño en una cuadrícula definida por la intersección de bandas de fila y columna. Es como una celda de una tabla o una celda en una hoja de cálculo. Si define una

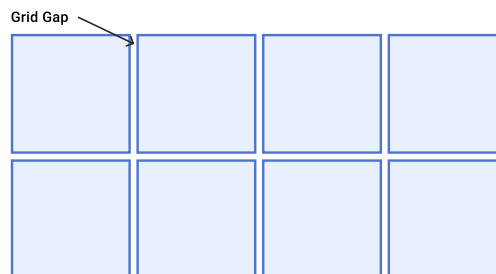
cuadrícula y no coloca ninguno de los elementos, se colocarán automáticamente un elemento en cada celda de cuadrícula definida.



- **Área de cuadrícula:** Varias celdas de la cuadrícula juntas. Las áreas de cuadrícula se crean haciendo que un elemento se extienda por varias bandas.



- **Brechas:** Una cuneta o callejón entre vías. A efectos de dimensionamiento, estos actúan como una banda normal. No puede colocar contenido en una brecha, pero puede distribuir elementos de cuadrícula a través de él.



Las propiedades clave de css Grid para definir la cuadrícula son:

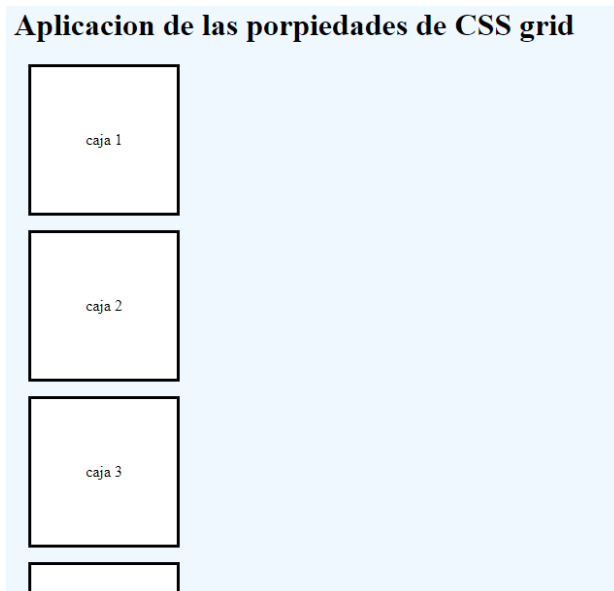
- **Grid-template-column:** Esta propiedad especifica el tamaño y la ubicación de un elemento de la cuadrícula dentro de una columna de la cuadrícula contribuyendo con una línea, un intervalo o nada (automático) a su ubicación en la cuadrícula, especificando así el borde de inicio y fin en línea de su área de cuadrícula. Dentro de la propiedad se pueden declarar su posición si al inicio o al final, con los comandos `Grid-column-end` y `Grid-column-start` respectivamente.
- **Grid-template-row:** es una pista horizontal en un diseño de cuadrícula CSS, es el espacio entre dos líneas de cuadrícula horizontales. Se define por la propiedad `grid-template-rows`. Además, se pueden crear filas en la cuadrícula implícita cuando los elementos se colocan fuera de las filas creadas en la cuadrícula específica. Estas filas serán de tamaño automático por defecto, o pueden tener un tamaño especificado con la propiedad `Grid-auto-row`.

El resto de las propiedades funcionan de la misma manera que el Flexbox por lo que aplicarlas resultara sencillo. Aplicando las propiedades en un ejemplo práctico; se inicia creando el concepto de casillas, que para este ejemplo se han denominado cajas.

## Aplicacion de las propiedades de CSS grid

caja 1  
caja 2  
caja 3  
caja 4  
caja 5  
caja 6  
caja 7  
caja 8  
caja 9

Al aplicarle los estilos correspondientes para identificar cada caja obtenemos

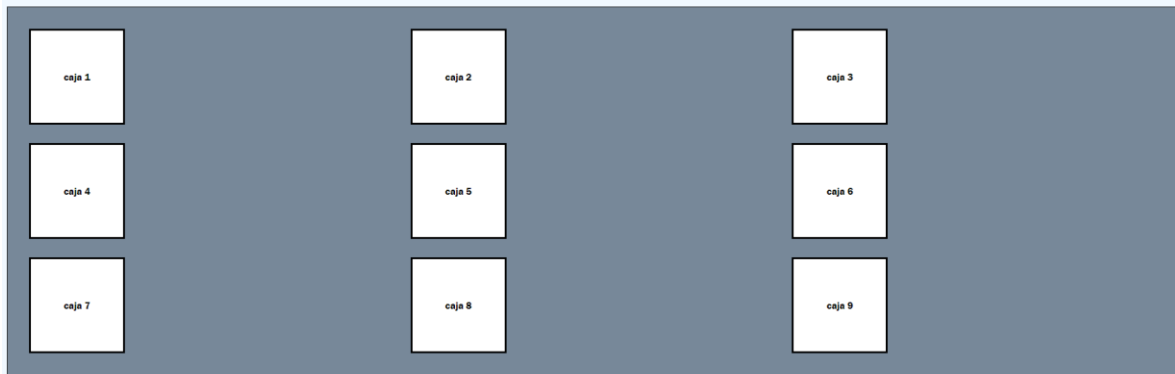


Al destacar el container y aplicar la propiedad display: Grid, se obtiene el siguiente resultado



Al declarar la propiedad Grid-template-column y establecer 3 columnas, con ancho automático, las cajas se ajustarán automáticamente hacia la izquierda, ya que anteriormente se han puesto márgenes y tamaños predeterminados a las cajas

#### Aplicacion de las propiedades de CSS grid



Al quitarles el tamaño a las cajas, pasaran a ocupar la totalidad del espacio dado

#### Aplicacion de las propiedades de CSS grid

caja 1	caja 2	caja 3
caja 4	caja 5	caja 6
caja 7	caja 8	caja 9

También se pueden asignar la cantidad de filas, su tamaño y margen con la propiedad Grid-template-row, quedaría de la siguiente forma:

#### Aplicacion de las propiedades de CSS grid

caja 1	caja 2	caja 3
caja 4	caja 5	caja 6
caja 7	caja 8	caja 9

Al cambiarle el largo a las filas, dando prioridad a la primera fila, daría como resultado:

#### Aplicacion de las propiedades de CSS grid

caja 1	caja 2	caja 3
caja 4	caja 5	caja 6
caja 7	caja 8	caja 9

Si a esto se le aplica el comando Justify-content, y se establece en center, se observa que las filas y columnas puestas en auto se acoplaran al tamaño disponible dentro del contenedor.

Aplicacion de las porpiedades de CSS grid

caja 1	caja 2	caja 3
caja 4	caja 5	caja 6
caja 7	caja 8	caja 9

Para finalizar, se le puede asignar el tamaño de la columna o fila a cada caja por individual, para la caja 1, 5 y 9 se da la totalidad de la columna, por medio de las líneas de cuadrícula.

Aplicacion de las porpiedades de CSS grid

caja 1		
caja 2	caja 3	caja 4
caja 5		
caja 6	caja 7	caja 8
caja 9		