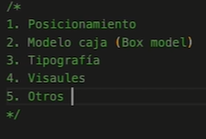
**CURSO DE MEDIA QUERIES**

**\*En los media queries:** min-width = desde max-width = hasta

**\*REGLAS DE CSS**

Orden para estilizar y buenas practica de estilizar convención



**\*UNA PAGINA WEB SOLO PUEDE TENER UN H1 POR TEMAS DE SEO**

\*iso de display: flex



\***Evitar que se desbor de el contenido de un contenedor**

Display:flex;

Flex-wrap: wrap

**\*MEDIDA PARA LA PANTALLA EN EL MODO DESARROLLADOR**

320\*640 EN MI CELULAR S9 🡪320 ES LA MEDIDA MAS PEQUEÑA EN LOS CELULARES

PARA CENTRAR UNA IMAGEN CON CONTENDOR DISPLAY FLEX SERIA CON ALING-SELF: CENTER

**\*USO DEL linear-gradient()**



Los valores van separados por una coma

Primer valor. Son los grados de difuminarían entre los dos colores principales

El segundo valor es es el porcentaje del primer color y el tercer el porcentaje del segundo valor}

\***botón flotante**: se le conoce al botón que esta dentro de dos contenedores y que tiene

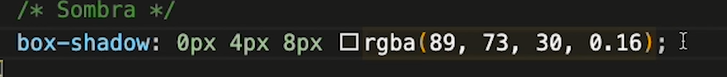
position: absolute

\*en position: absolute

Podemos utilizar top, left, right , bottom

\***Uso de sombras:**

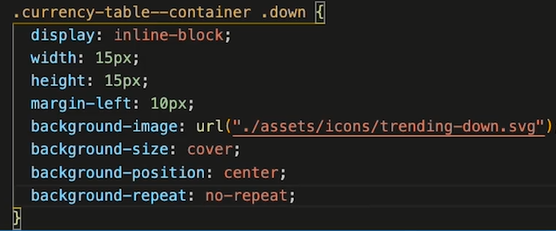
Box-shadow



**\*imagen de fondo en un span “li – a o cual quier contenedor”**  debe tener un display (inline por defecto)

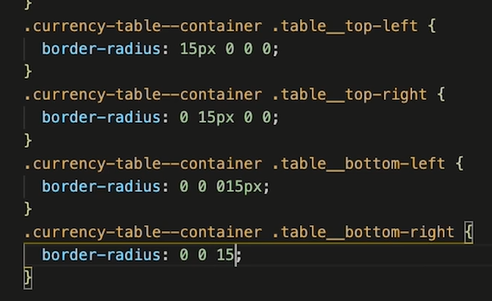
Se le debe poner display : inline-block

El .down fue el nombre selector de clase que se le puso a un span



\***texto en negrita html:** con la etiqueta <b>

**\*Border-radius en las tablas:**



**Superior izquierda: 1 valor, superior derecha: 2 valor inferior izquierda: 4 valor, inferior derecha: 3 valor**

**\*padding**

 **primer valor: arriba y abajo / segundo valor: derecha , izquierda**

**\*nota cuando la imagen que necesitamos ubicar esta sola, no usamos el display: inline\*block**

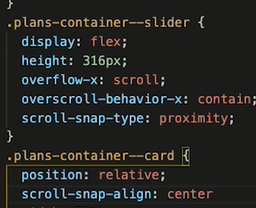
**\*para que el contenido no se salga del contenedor padre:** si le damos un width y un height a un componente estamos limitando su crecimiento o reducción y si el contenido es más grande que el contenedor padre entonces se desbordara el componente LO SOLUCIONAMOS CON DARLE UN min-width y min-height

**\*imagen de fondo para una sección completa**  lo mismo que para los otros

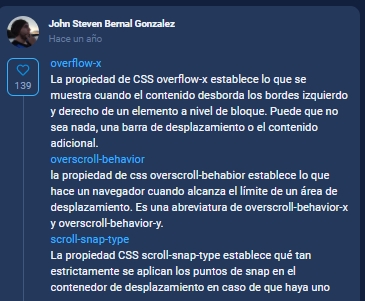
**\*generar automáticamente columnas a medida que se aumente el tamaño de la pantalla.**



**\*scroll horizontal**



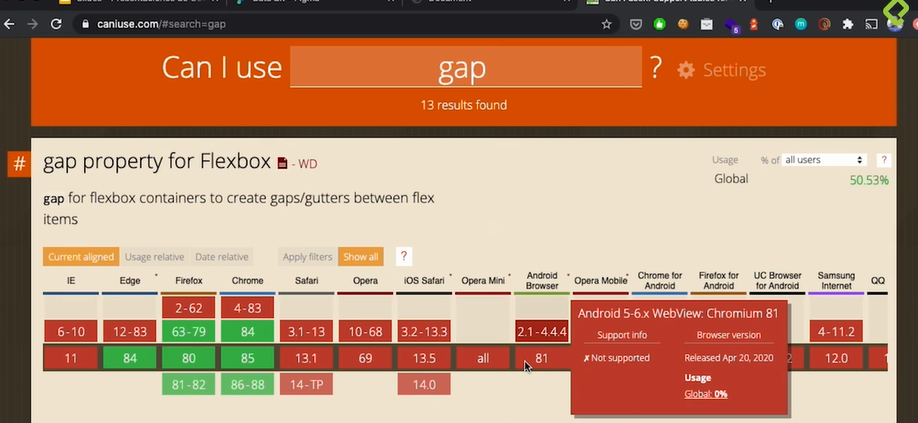




El primer contenedor es el principal el segundo son los articles

**\*Can i use**

Para verificar si los estilos que le ponemos a nuestra página están implementados en los navegadores

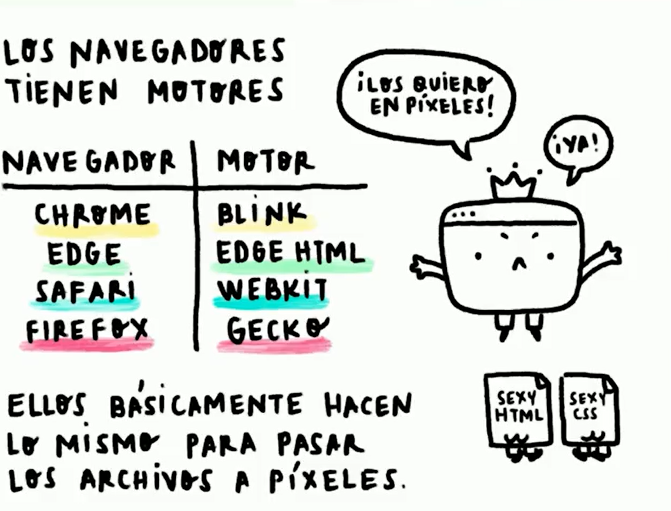


\*como verificar si nuestro sitio web cumple con ciertas características que nos llevan a ser mejores desarroladores

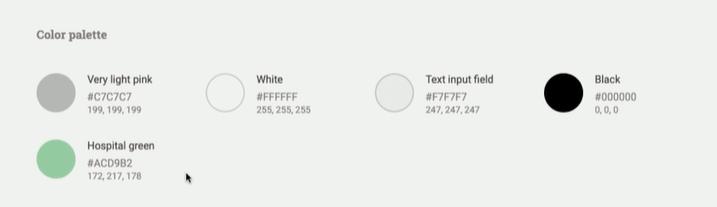


CURSO DE FRONTEND-DEVELOPER

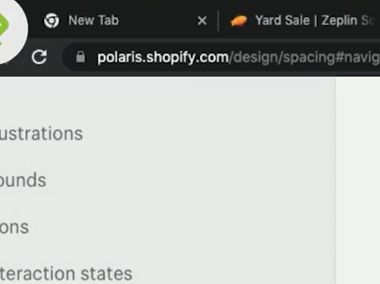
Motores de renderizado: interpretes entre lenguaje de programación y navegador



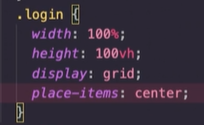




**\*SISTEMAS DE DISEÑO:** SHOPIFY



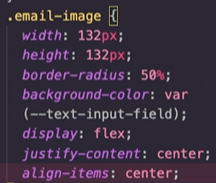
**\*centar imágenes con css grid**



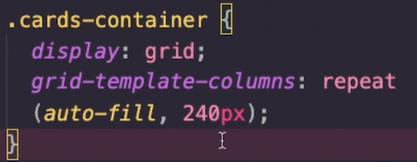
**\*PARA QUE EL ELEMNTO NO APAREZCA EN UN BREAKPOINT: DESKTOP / MOVIL**

**display : none**

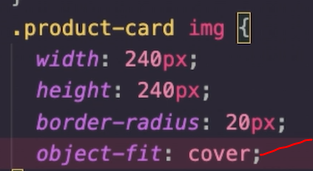
**\*OTRA FORMA DE CENTRAR UNA IMAGEN:**  ES UN DIV CON UNA IMG DENTRO



\***generar columnas automáticamente a medida que se aumente el tamaño de la pantalla y con un tamaño determinado ( 240px)**



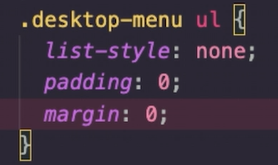
**\*hacer que la imagen se vea bien independientemente del tamaño del contenedor**



**\*ponerle 35px a una imagen se conoce como número random**

**\*hacer que el primer elemento de algo como ene este ejemplo sería el primer elemento p que este dentro de div de la clase procuct-info va a tener los siguiente**

**\*NOTA: A LOS UL SE LES PUEDE QUITAR EL MARGIN Y EL PADDING**

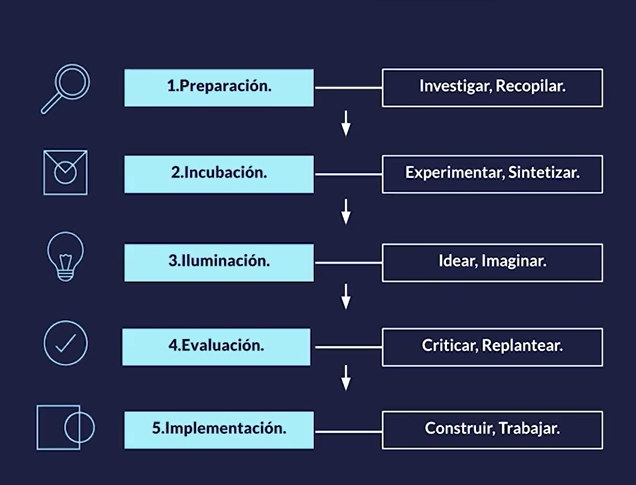


**NOTA: A un elemento inline como ul li a no se le aplica margin**

**\*para quitar elementos al momento de aplicar media queries se utiliza display NONE Y viceversa seria con display block**

CURSO DE DISEÑOS PARA DEVELOPERS

PROCESO CREATIVO

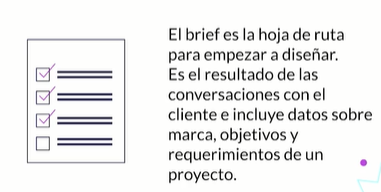


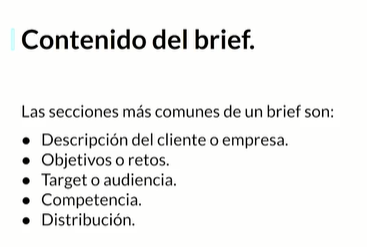
**\*CONCEPTOS BASICOS**

**BALANCE / CONTRASTE / ALINEACION / PROXIMIDAD /REPETICION / ESPACIO**

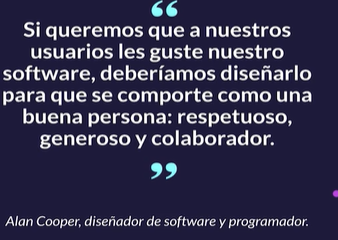
**\*DISEÑO RESPONSIVO**

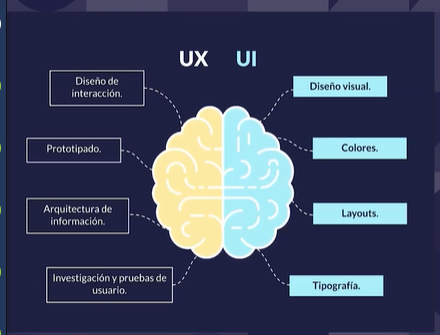
**\*PRIMEROS PASOS A LA HORA DE CREAR UN SITIO WEB**





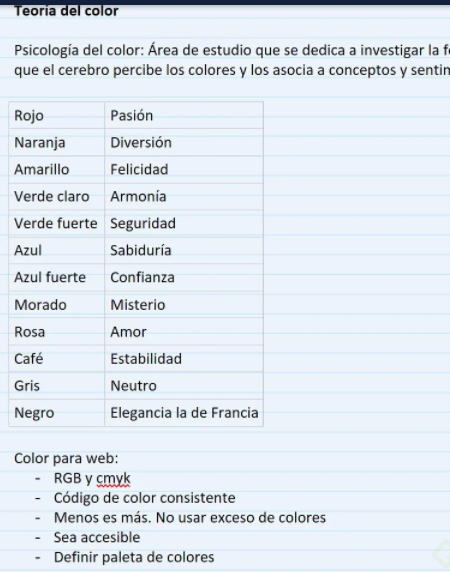
**\*DISEÑO UX**







**\*TEORIA DEL COLOR**

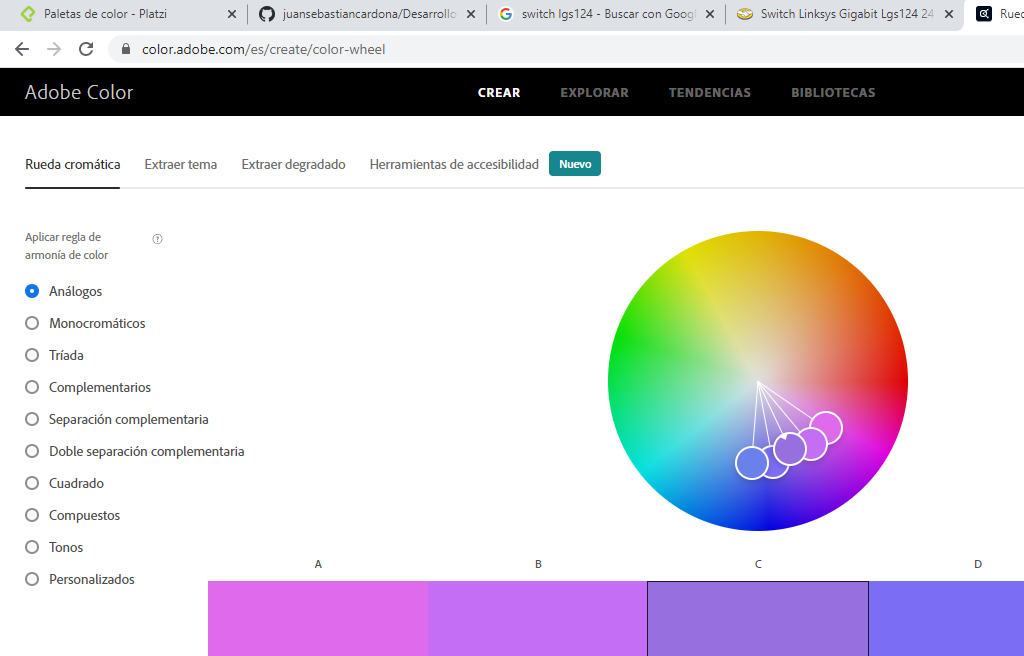
 



**Combinación de colores o paletas**



**GENERADOR DE PALETA DE COLORES**



**TIPOGRAFIA**



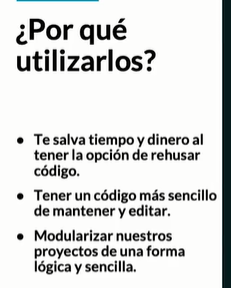
No letras cursivas en responsive design por que la letra se encoje

**\*LAYOUT Y SISTEMA DE GRILLAS**

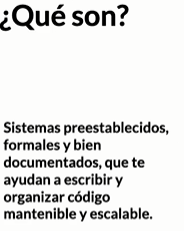
**Grid**

**CURSO DE PREPROCESADORES**

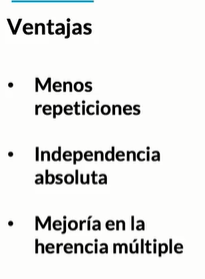








**Ventajas de BEM**



**GUIA PARA LA CREACION Y MANTENIMIENTO DE CODIGO:**

**NOMBRE DE ARCHIVOS EN PLURAL**

**CLASES EN SINGULAR Y EN MINUSCULA**

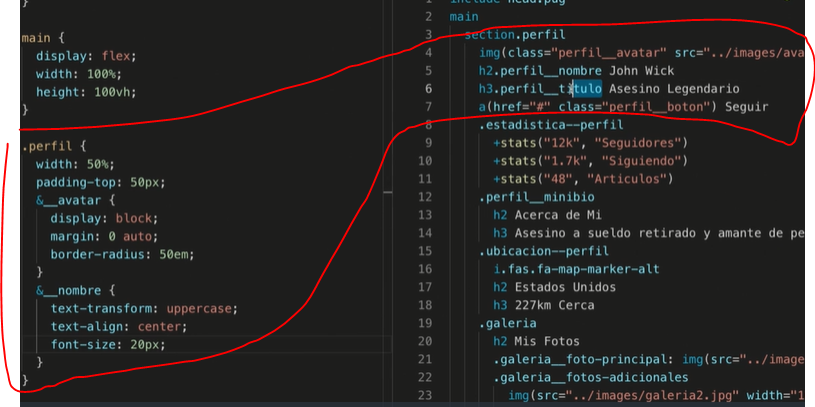




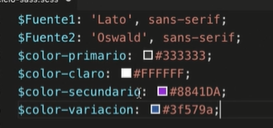
**\*SASS**

Extensión del archivo **.scss**

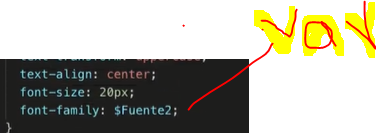
**\*ANIDACION :** la sección con clase perfil hace la anidación con sus elementos internos que son \_\_avatar\_\_nombre / \_\_titulo / \_\_boton



**\*Variables**



**Llamar variable**

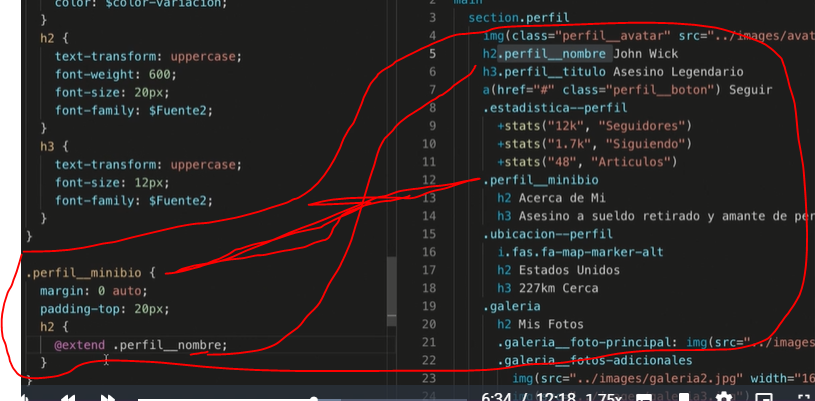


**\*import y extends**

**El extend**

**sirve para “clonar ” los estilos de un elemento**

**Ejemplo vamos a clonar los estios del h2 que esta en elemento de clase perfil al h2 que esta en el lemento de clase perfil\_\_minibio**



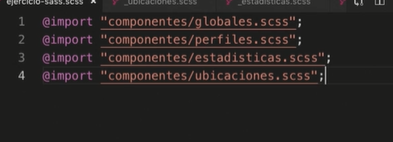
**El import**

Sirve para modularizar nuestro archivo .scss. (estilos.scss -> y se nombran los archivos se nombran con raya al piso para que no se compilen o para que prepos no cree un csss por cada uno)

Para cada elemento creamos un archivo sass por cada contenedor y en el archivo principal los llamamos con el import

Siempre llamamos el archivo que tiene las configuraciones (variables / mixin estilos generales) primero

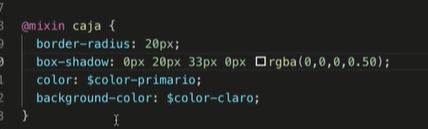
**NOTA**: HAY QUE IMPORTAR LOS ARCHIVOS EN EL ORDEN EN QUE ESTA ESTRUCTURADO EL HTML



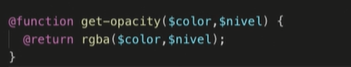
**\*Mixins:**

**Parte de código reutilizable entodo nuestro proyecto como cundo algunos elementos tienen los mismos estilos como cajas articles**

**Aquí le estamos colocando el color al fondo del conteneor a la letra, borde y sombreado**



**\*funciones**

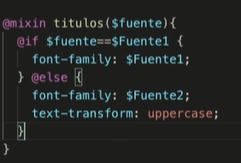


**Llamar la función ny colocando parametreos**



**\*condicionales y loops**





**Llamar mixin**

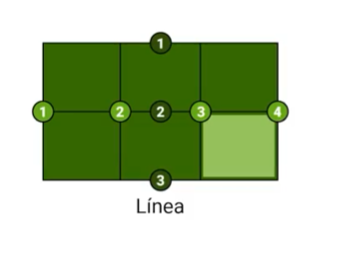


CURSO BASICO DE CSS GRID

Css nace de la necesidad de hacer mas dinamico la forma en hacer estructuiras como tablas y layouts utilizando un sistema de grilla

**Contenedor**: el elemento que se convierte en una grilla y los elementos serian los ítems

**Línea** elementos que limitan los elementos de la grilla



Celda

unidad ininima ded u

Grupos de celdas

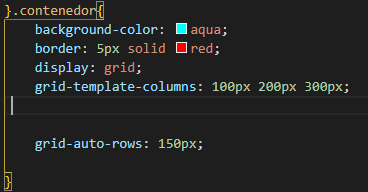
Tracks en una mis fila o clumna

Áreas varias filas o varias columnas

\*propiedades del contenedor



\*grill a de 3 columnas con dos filas que se crean automáticamente con un alto de especifico

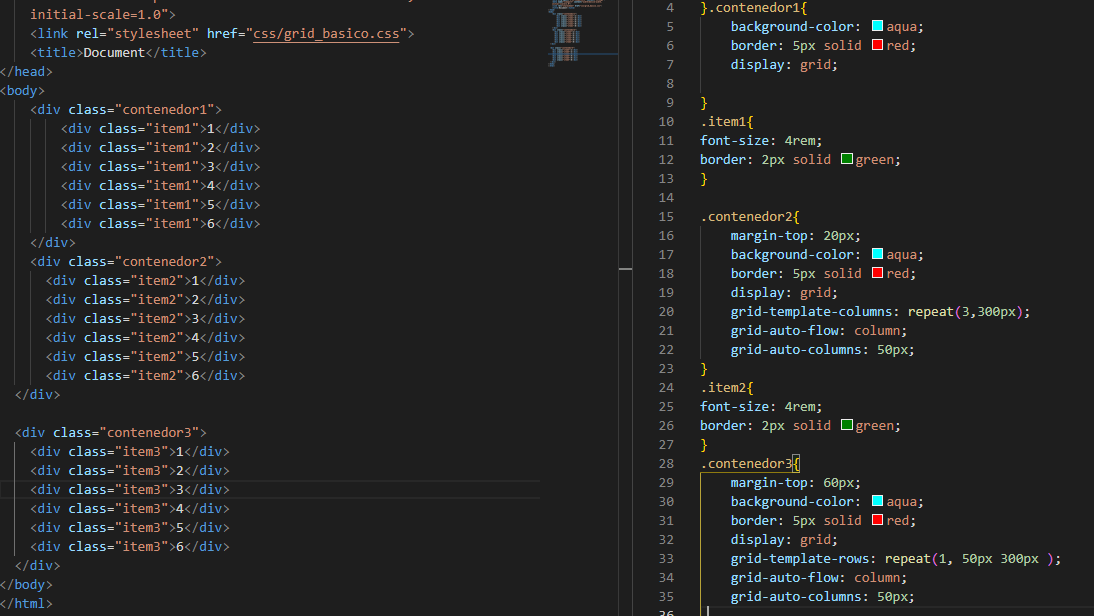


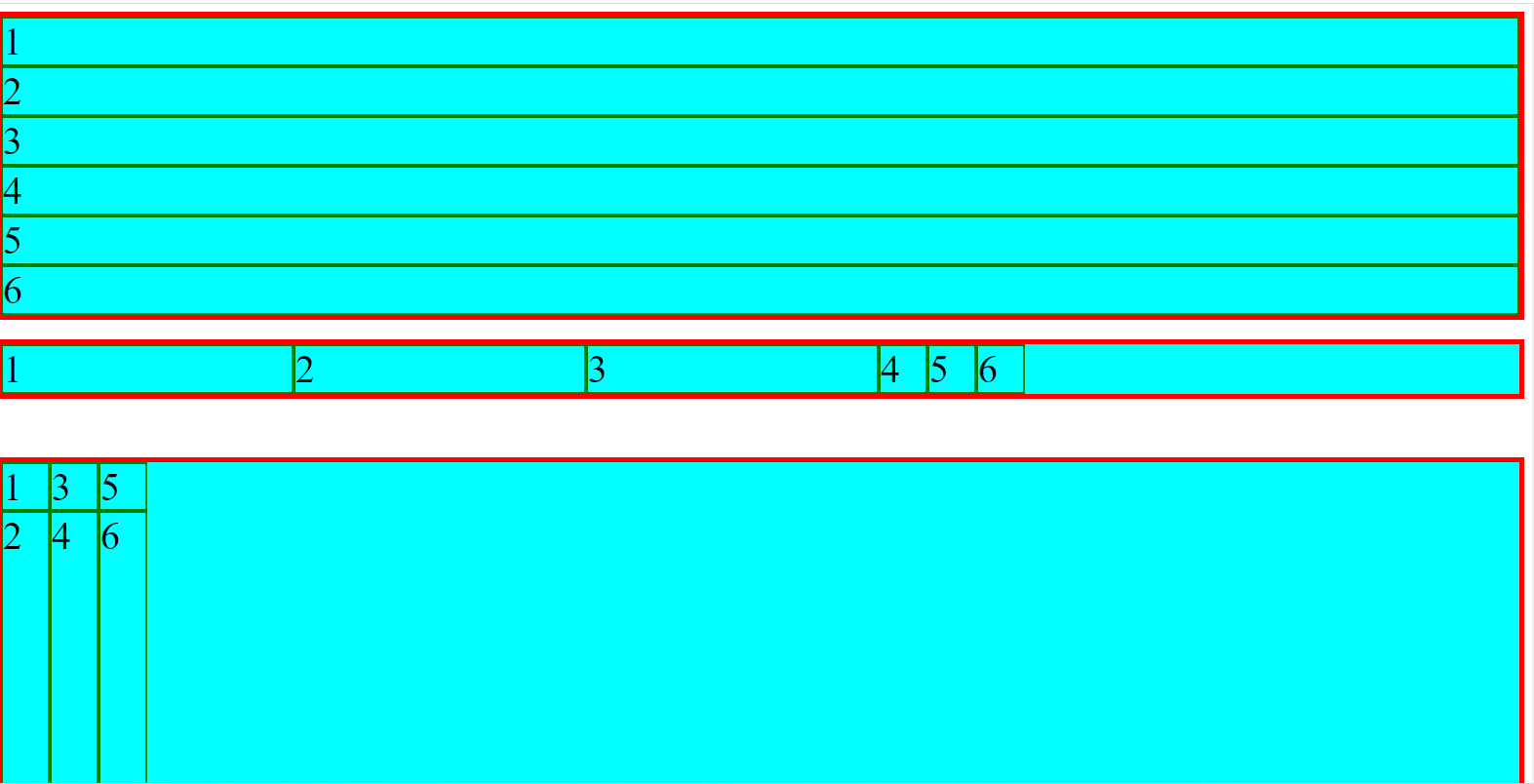
\*NOTA al llenar la grilla lo hace siempre por las columnas

GRID-AUTO-FLOW ESPARA CREAR FILAS O COLUMNAS ATOMATICA/

Si queremos que genere nuevas columnas serial con el gri-auto-flow: columns

\*para hacer que la grila se llene por la posición de las filas serial con el gri-auto-flow:row





**\*PARA PONER ESPACIOS EN NUESTRAS GRILLAS**

CONLA PROPIEDAD gap

**Row-gap** : para poner un espacio filas determinado

entre las filas

**Columns-gap:** para poner un espacio filas determinado

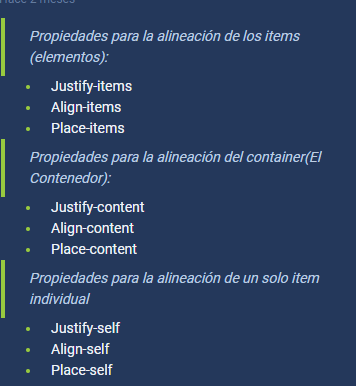
entre las columnas

Gap: para ambas uniformemente

Reto:

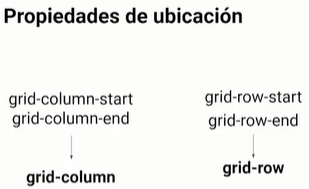


**\*PROPIEDADES DE ALINEACION**



**\*PROPIEDADES DE UBICACIÓN**

**COMO CAMBIAR LA UBICACIÓN DE LOS ELEMENTOS**



**CLASE 5**