# **CSS** parte II

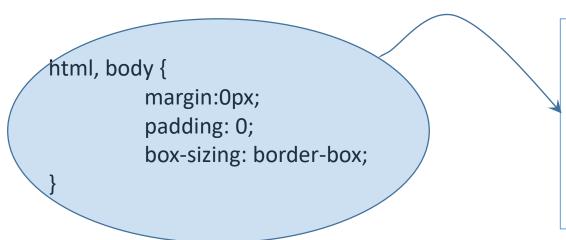
Posicionamientos absolutos, relativos y fijos

Cajas flexibles

Menues

Maquetación y elementos RWD (Responsive web design)

#### Posicionamiento relativo de elementos dentro del documento



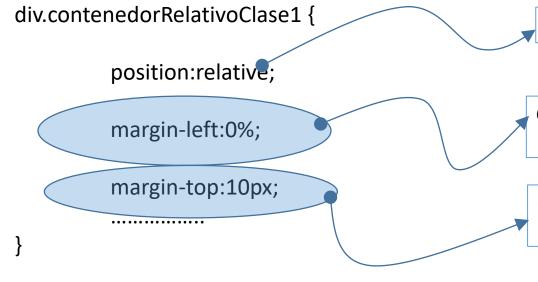
Configuración default o de base valida para tanto el documento como para el body.

Nigún elemento en su interior deja márgenes entre si. Ningún elemento contenedor guarda separación respecto a su

contenido.

Cualquier padding o ancho de borde no afecta el ancho y el alto definido para un elemento.

Valores por default que se me ocurre especificar para toda mi librería de estilos.

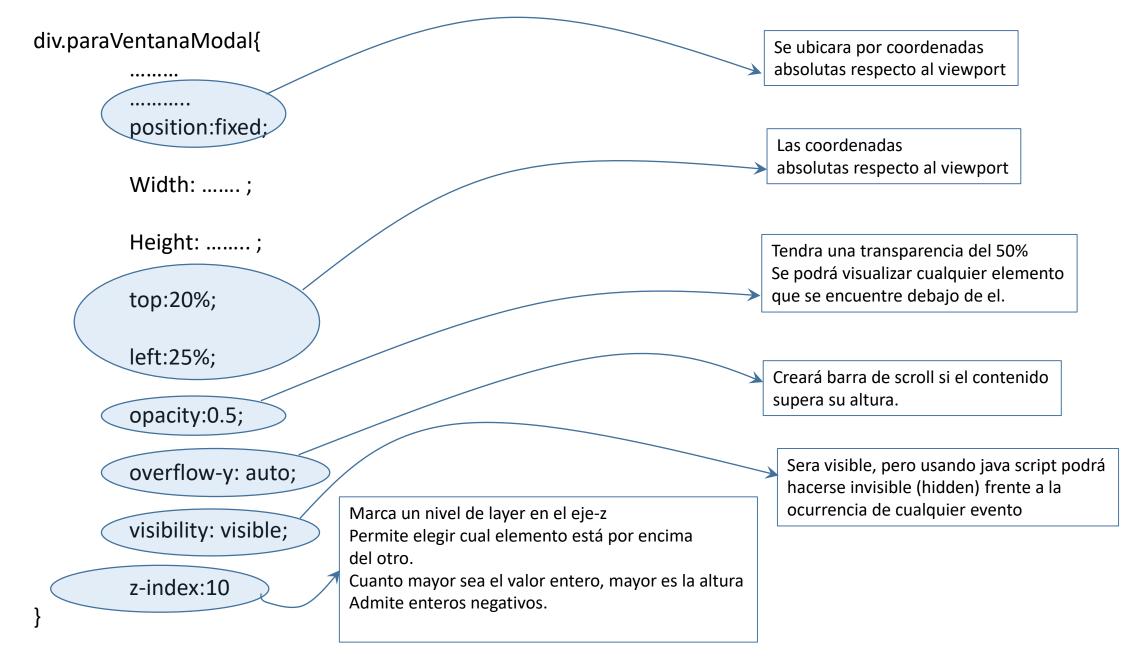


Posicionamiento default. Si no se especifica el navegador lo asume.

Coordenada en x del elemento respecto al borde del contenedor padre.

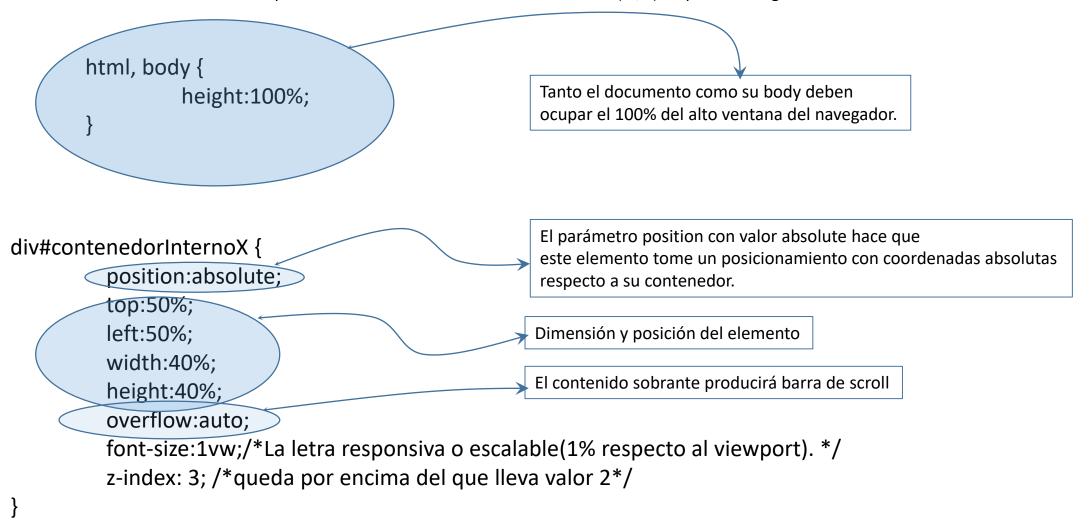
Coordenada en y del elemento respecto al borde inferior del div anterior

#### Estilo para una ventana modal que no se mueve con el scroll del contenido



#### Posicionamiento absoluto de elementos

Se trata de posicionar elementos html de manera absoluta respecto a su elemento contenedor, tomando como origen de coordenadas la esquina superior izquierda del mismo. En el ejercicio se pretende obtener una vista del documento sin barras de scroll de ningún tipo. Todo elemento con posicionamiento absoluto tiene su ubicación (Xi,Yi) respecto al origen de coordenadas de su contenedor.



# **Cajas Flexibles**

display: flex;

justify-content: center;

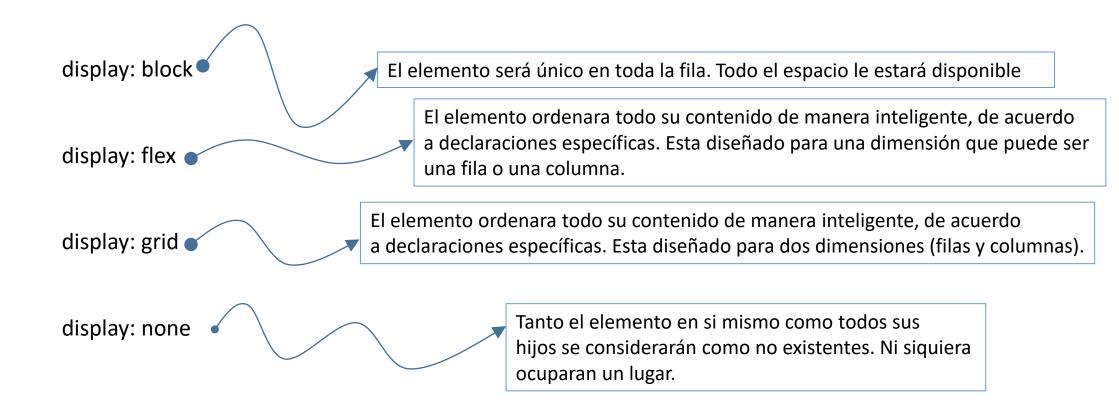
align-items: center;

Estos parámetros de estilo son para que todos los elementos que conforman el contenido interno dentro de una caja queden centrados dentro de la misma tanto en los ejes "x" (justify-content) como "y" (align-items) Todos lo elementos dentro de la caja serán distribuidos dentro de la misma de una manera inteligente de acuerdo a los parámetros usados.

Ver Flex boxes.

## Propiedad display

La propiedad display en CSS especifica si un elemento es ubicado como único en una línea (valor block) o como un elemento con todo su contenido interior (elementos hijos) ordenado automáticamente de acuerdo a parámetros específicos (valores flex o grid).



#### El modelo de caja flexible

display: flex;

flex-direction: row;

Justify-content:center;

Permite definir el eje principal del flexbox. Un valor puede ser row y otro column. Siempre el modelo flexbox trabaja en un eje

Permite centrar elementos al medio del eje principal del flexbox Permite flexibilizar el contenido reubicando y redimensionando elementos dentro del mismo en forma automática según distintos parámetros que pueden ser pasados.

justify-content: space-around;

Permite centrar elementos al medio en el eje principal pero espaciándolos entre ellos.

Align-items:center

Permite centrar elementos al medio en el eje secundario que siempre es ortogonal respecto al principal.

Flex-wrap:wrap;

Permite administrar bloques internos de manera de enviarlos automáticamente a la siguiente línea en caso de disminuir el ancho del viewport del dispositivo

### Menues desplegables

El elemento nav con id="menú" será el contenedor de todo el menu

Esto se lee así:

Toda lista ul dentro del menú principal no lleva viñetas

#menu ul li { background-color:#2e518b;

list-style:none;

#menu)ul {

Todo elemento li dentro de un ul dentro del menú principal son los elementos de opción de dicho menú.

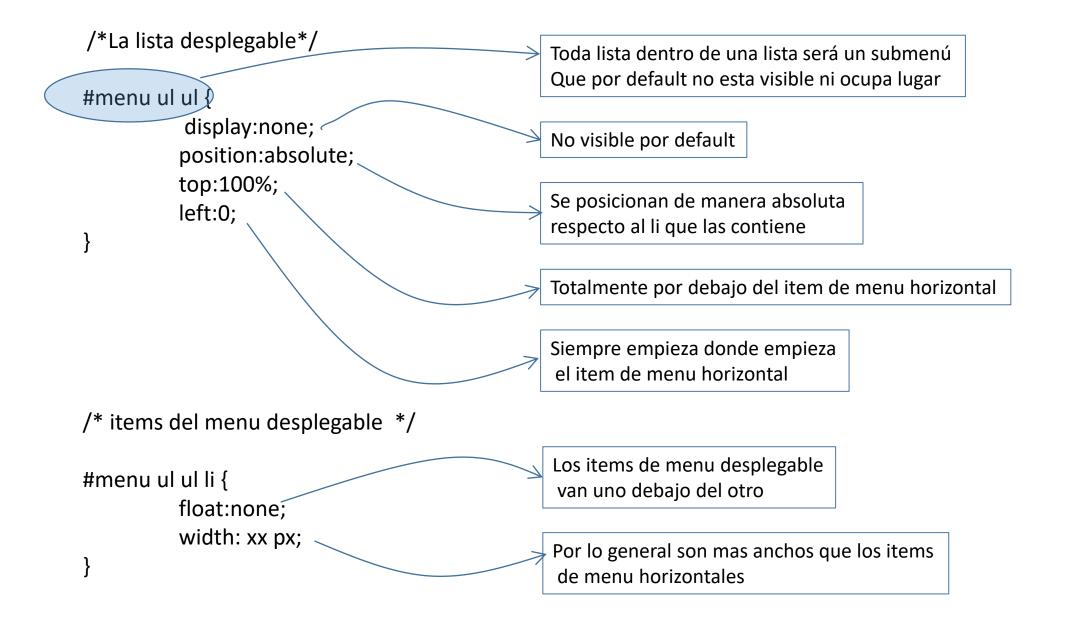
En este caso llevaran un color de fondo determinado.

 Los enlaces <a> dentro de un ul dentro del menú serán siempre visibles y ocuparán todo el ancho del contenedor padre (li). O sea que los anchors siempre estarán dentro de un li y lo ocuparan por completo.

#menu ul li {
 position:relative;
 float:left;

Como las li del menú principal pueden contener listas de submenues desplegables por dentro con posicionamiento absoluto es que tengo que usar position relative en este contenedor li que actuará como padre.

Los ítems del menu principal se acomodan en fila.



```
#menu ul ul a {

padding:10px;
}
```

Esto se lee de la siguiente manera:

Todo enlace <a> dentro de una lista que a su vez esta dentro de otra lista que a su vez está dentro del menú principal

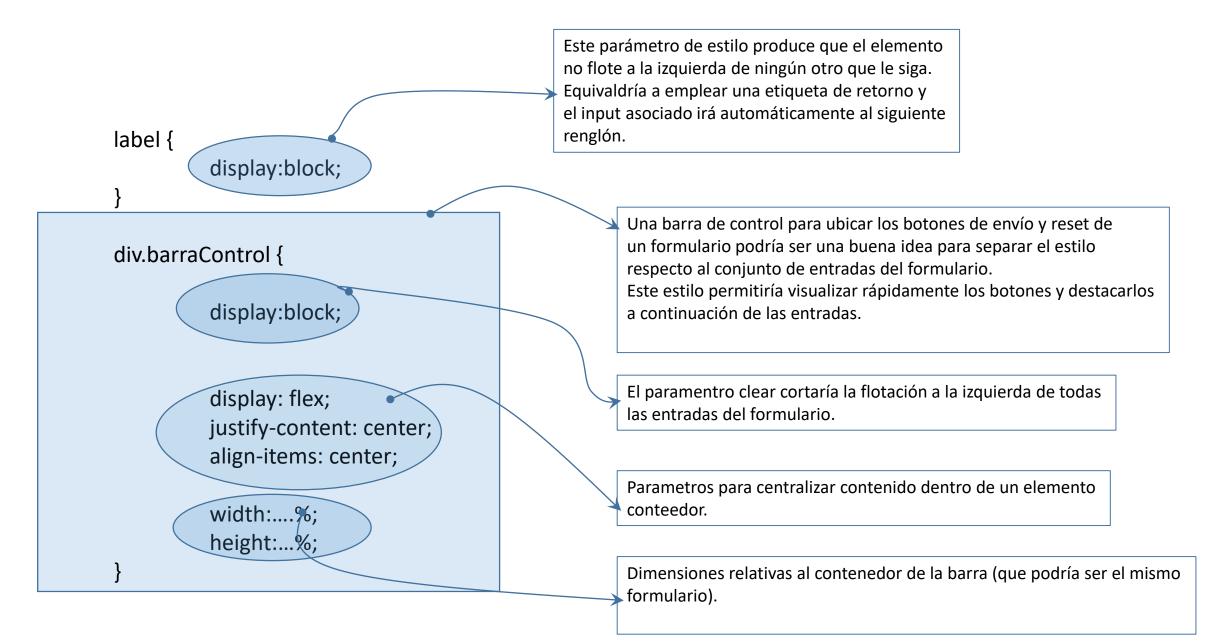
Para que queden un poco indentados respecto a los items de menu horizontal

Esto se lee de la siguiente manera:

Toda lista (ul) dentro de un li en estado de mouse over y a su vez dentro de una lista (ul) del menú principal

La lista desplegable (ul) se hace visible y ocupando todo el contenedor padre (li) cuando dicho padre tiene el mouse arriba

#### Detalles de estilo para algunos elementos en un formulario RWD (Responsive Web Design)

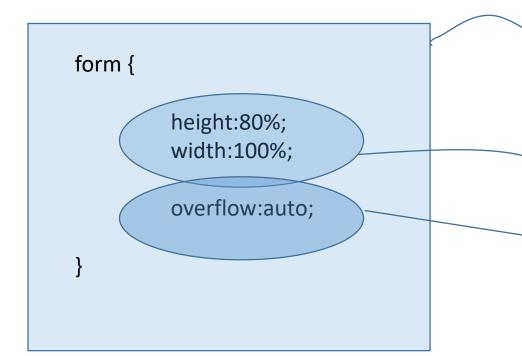


```
.boton {
    width:10%;
    height:80%;
    background-color:#AAAAAA;
    cursor:pointer;
}
```

Esta clase define un estilo para cualquier botón del documento.

Las dimensiones son relativas al contenedor.

Recordar que se debe establecer toda la cadena completa de alturas de todos los elementos que se contienen. (html – body – contenedor general – contenedor del formulario – form – barraControl – botón.



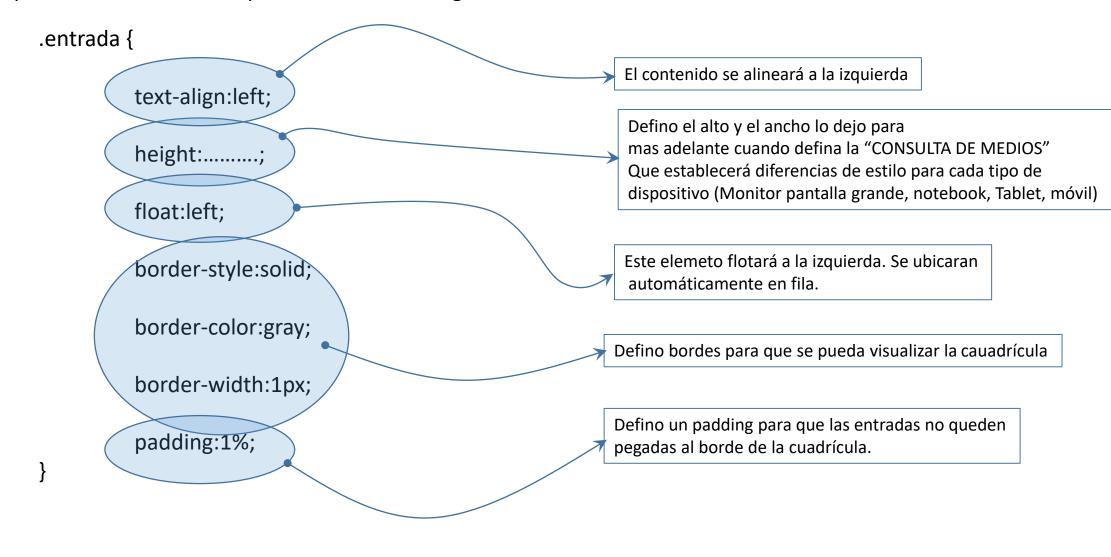
Estilo valido para cualquier formulario que se encuentre en el documento.

Las dimensiones del div serán relativas a su contenedor.

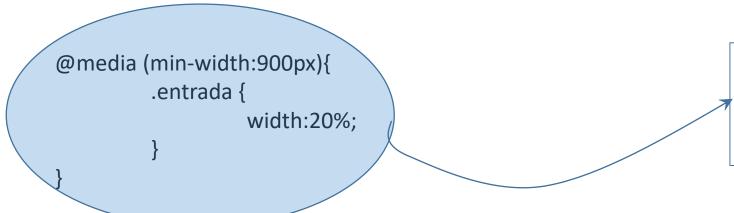
Aparecerá la barra de scroling si es necesario.

#### Las entradas son la agrupación de label + input.

El estilo para la clase entrada se puede definir de la siguiente manera.



#### Consulta de medios (valida para otorgar la responsividad o adaptación al medio)



Si el ancho del view port del dispositivo es mayor a 900px Entonces....

Estilo válido para dispositivos grandes, mayores a 900 pixels. Resultarán 5 entradas por fila.

```
/*Si el dispositivo tiene un ancho mayor a A y menor que B, entonces .. */
```

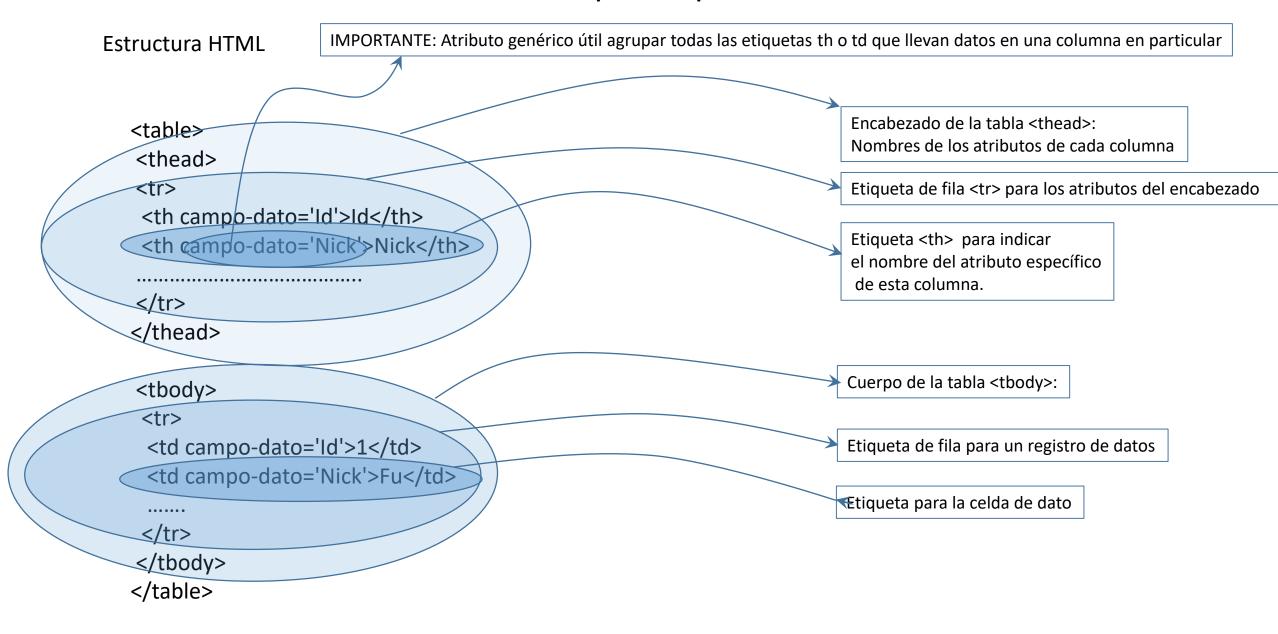
Si el ancho del view port del dispositivo está entre 700px y 900px

Entonces....

Estilo válido para dispositivos medios.

Resultarán 4 entradas por fila

#### Tabla con estilo para incorporar datos



#### Selector para los atributos genéricos de determinado valor

```
[campo-dato='Id'] {
   width: 5%;
}
```

Nuevo selector para agrupar estilos: En este caso, todos los elementos con atributo campo-dato con valor 'id' tendrán tal ancho En este caso todas las etiquetas HTML que tengan un atributo campo-dato con el valor "id" Tomarán este estilo (En este caso un ancho de columna del 5 % de toda la fila.

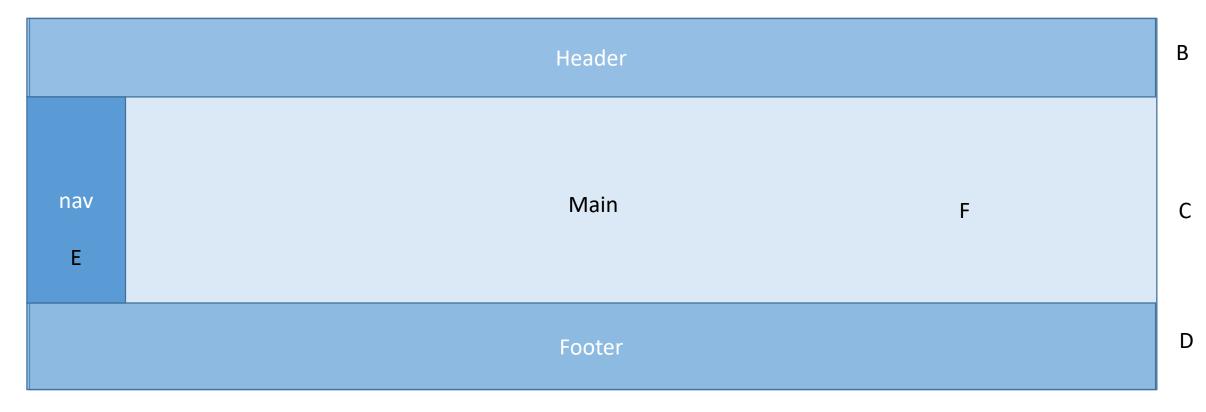
#### En el caso de querer controlar estilo de un elemento con media query:

En este caso se especifica que todas las etiquetas HTML que lleven el atributo campo-dato='Skype' no se visualizarán ni ocuparán lugar en pantalla si la media corresponde a un dispositivo con menos de 1200 pixels de ancho.

#### En el caso de querer controlar estilo de filas impares :

```
tbody tr:nth-child(odd) {
 background: rgba(0,0,0,.2);
}
```

Selector que permite definir estilo para las filas impares de tbody



- 1) El Contenedor "A" de la maqueta contiene todos los elementos (puede ser el body del documento)
- 2) "A" Se divide en tres contenedores al 100% de ancho (header, nav + main, footer) ("B","C","D") y al 33% del alto cada uno de ellos.
- 3) "C" se divide en dos contenedores al 100% de alto (nav al 10% del ancho, main al 90% del ancho) ("E","F")

Aclaración: nav y main deben flotar a la izquierda (float:left) y seguido irá un contenedor de corte (clear:both) Alternativa para evitar el contenedor de corte es que C tome un parámetro display:block

Dentro de la etiqueta main podría incluirse un iframe con nombre (name) para poder usar como target en los anchor's del menú.

#### Corrimiento y rotación de elementos

<img src="globosFT.png" style="transform: translateX(-50%);" />

Corrimiento de una imagen en x hacia la izquierda una distancia equivalente al 50% del contenedor de la misma

<img src="globosFT.png" style="transform: rotateZ(10deg);" />

Rotación de una imagen en 10 grados respecto al eje perpendicular al plano de la misma.