

Ejercicio de Diseño en 2 columnas:

- Os facilito el código de este ejercicio para que veáis cómo se suele organizar el contenido de una página web.
- El código tiene ya comentarios para que se entienda mejor lo que se debe de hacer.
- En este caso queremos crear una estructura en 2 bloques, uno al lado de otro => imprescindible que sean elementos en **línea**, PERO, si utilizo **display:inline** NO se pueden utilizar las propiedades de redimensionado ni las de márgenes verticales (superior/inferior), de ahí que utilice **display:inline-block**, y así consigo poder utilizar esas propiedades, como elemento en bloque, y que se ponga uno al lado de otro.
- Uno de los ejercicios está hecho usando el display
- El otro, utilizando el float.
- A partir de ahí, id haciendo pruebas de cómo se va modificando la estructura a medida que cambiamos las propiedades
- **Recomendaciones generales:**
 - Utilizar las propiedades de redimensionado y las que definen una anchura máxima (max-width), para evitar que según la pantalla se haga “demasiado grande” o demasiado pequeña (min-width).
 - Aunque la etiqueta que contiene todos los elementos visibles de una página web es <body> no siempre interesa aplicar las dimensiones y color de fondo de la estructura principal desde es elemento, de ahí que hay incorporado, como elemento contenedor un <div> que encierra todo el contenido, cabecera y pie.

1) Pantallazo del código CSS utilizando float: left

```
/*contenedor div y dentro 2 secc, un articulo y un aside. como quiero colocarlos al lado,
uso display inline-block; para q el aside se sitúe arriba, usar el vertical-align.
Para indicar q es diseño centrado horizontalmente o utilizo la prop shorthand d margin o lo especifico individualm empleando las prop margin top, left y right.
Especificar tamaños de máxima y mínima anchura para cada bloque */
#wrapper {
    max-width:900px;
    min-width: 600px;
    margin-top:0px;
    margin-left:auto;
    margin-right:auto;
    background-color:black;
}
/* no es necesario convertirlo a elemento flotante, ya q ocupa todo el ancho definido */
#cabecera{
    height: 100px;
    background-color: #73A;
    max-width: 900px;
}
/* no es necesario convertirlo a elemento flotante, ya q ocupa todo el ancho definido. Para evitar problemas de posicionamiento con respecto
a los elementos flotantes superiores (aside y article), añadido el clear */
footer{
    background-color: #73A;
    height: 80px;
    clear: both;
    max-width: 900px;
}
nav {
    float:left;
    width:250px;
    height:300px;
    background-color: #A94;
    max-width: 250px;
    min-width: 100px;
}

article {
    width:640px;
    height:600px;
    float:left;
    margin-left: 10px;
    background-color: #E42;
    max-width: 640px;
    min-width: 300px;
}
```

```
<body>
  <div id="wrapper">
    <header id="cabecera">cabecera</header>
    <nav>barra navegación</nav>
    <article>contenidos página</article>
    <footer>footer </footer>
  </div>
</body>
```

Código Html:

2) Utilizando display: inline-block

```
/* ya no necesito eliminar el efecto de fluir con clear, pq no se está utilizando el posicionamiento flotante */
footer{
    background-color: #73A;
    max-width: 900px;
    height: 80px;
}
/* necesario añadir alineación vertical, para indicarlo (son elementos normales, no flotantes) */
article {
    width:600px;
    height:600px;
    background-color: #E42;
    display:inline-block;
    vertical-align:top;
    max-width: 600px;
    min-width: 300px;
}
nav {
    width:250px;
    height:300px;
    background-color: #A94;
    display:inline-block;
    vertical-align:top;
    max-width: 250px;
    min-width: 100px;
}
```

(el resto de elementos son iguales q en el caso anterior, igual q el código Html)