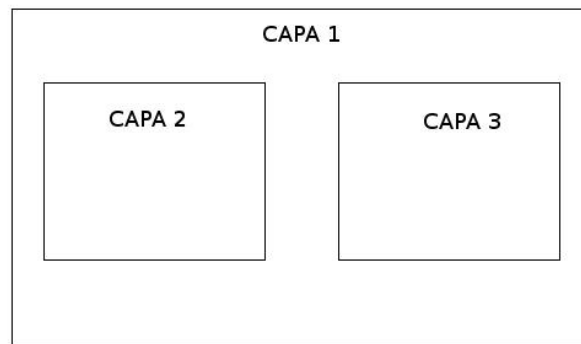


## Guía: Pasos para maquetar una página web con DIVs (capas o layouts)

**Maquetar una página web es pasar el diseño a código HTML**, poniendo cada cosa en su lugar (una cabecera, un menú, etc.). Hasta hace unos años la única manera de maquetar una página web era mediante tablas HTML (<table>), pero esto tiene muchas desventajas y limitaciones, por eso la técnica de maquetación fue evolucionando con los años hasta llegar al punto donde no se usan tablas, si no **capas** (DIVs) a las que se le dan formato mediante CSS.

**Las capas, layouts o divs son la misma cosa** con distinto nombre, para tener un concepto mental más fuerte de lo que son, podemos imaginarlos como ***contenedores o bloques donde podemos meter lo que queramos dentro*** (imágenes, texto, animaciones, otro bloque, o todo al mismo tiempo) a los que se le asigna un ancho, alto y posición, de esta manera se van a ir posicionando consiguiendo la estructura que queremos.

Veamos un ejemplo gráfico:



Como podemos ver, tenemos 3 capas estructuradas de la siguiente manera:

- **Capa 1:** es la capa principal y contenedora
- **Capa 2:** capa dentro de la capa contenedora 1 y alineada a la izquierda (float:left;)
- **Capa 3:** igual que la capa 2, solo que tiene un margen con respecto a esta ultima (float:left; margin: 10px;).

### 1. Creando nuestro primer DIV:

La forma de crear una capa DIV es así: <div></div>, recuerda que todo contenido visible de la página debe ir entre las etiquetas <body></body>, con el código HTML completo quedaría así:

```
<html>
  <head>
    <title>Curso de maquetacion CSS</title>
  </head>
  <body>
    <div>¡Esta es mi primer capa!</div>
  </body>
</html>
```

## 2. Dar formato a un DIV

Para dar formato a un DIV tenemos que identificarlo de alguna forma, para esto existe **el atributo ID**, en el pondremos el nombre del DIV para luego llamarlo desde la hoja de estilos, la forma de escribirlo es así:

```
<div id="capa1">¡Esta es mi primer capa!</div>
```

Como podemos ver, a esta capa le asignamos el nombre de "capa1", ahora solo nos falta abrir la hoja de estilos que creamos y llamarlo de esta manera:

```
#capa1{  
    background-color:green;  
}
```

Esto hará que el color de fondo *de esa capa* sea verde, en la siguiente imagen se puede apreciar:



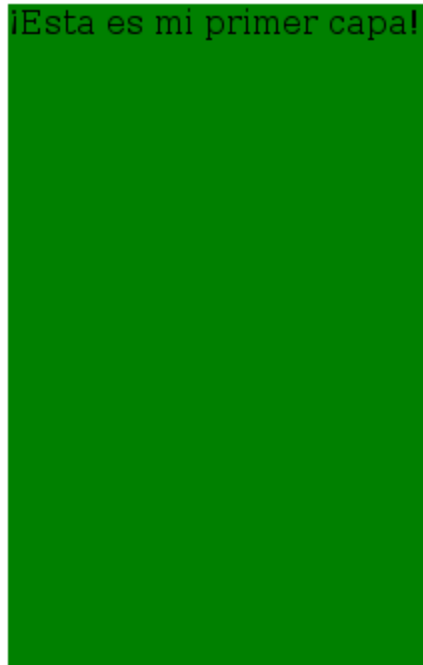
Como vemos, el color se estira, esto es porque no le definimos un ancho determinado, para hacer agregaremos el atributo **width** a #capa1 de la siguiente forma:

```
#capa1{  
    width:210px;  
    background-color:green;  
}
```

Se verá el fondo verde en un área mucho más corta como en este ejemplo, pero ¿qué pasa si queremos que la capa sea más alta?

Para ello agregaremos el atributo **height** con el valor en pixeles que queramos, de esta forma:

```
#capa1{  
    width:210px;  
    height:300px;  
    background-color:green;  
}
```



Con esto ya tendremos una especie de rectángulo verde donde podremos agregar texto como queremos

### 3. Flotar y acomodar un DIV

La maquetación por divs CSS **se basa en "flotar" unas capas dentro de otras para conseguir la estructura que queremos**, para esto se utiliza la propiedad *float*.

Esta propiedad se utiliza para flotar los bloques, podemos utilizarla con los siguientes valores:

- **none:** no flota la capa.
- **left:** flota la capa hacia la izquierda.
- **right:** flota la capa hacia la derecha.

Siguiendo con el ejemplo de la lección anterior, vamos a crear otra capa #capa2, le vamos a aplicar otro color de fondo con el mismo alto y ancho pero vamos a flotar ambas capas hacia la izquierda, quedaría así:

```
#capa1{
  width:210px;
  height:300px;
  background-color:green;
  float:left;
}

#capa2{
  width:210px;
  height:300px;
  background-color:gray;
```

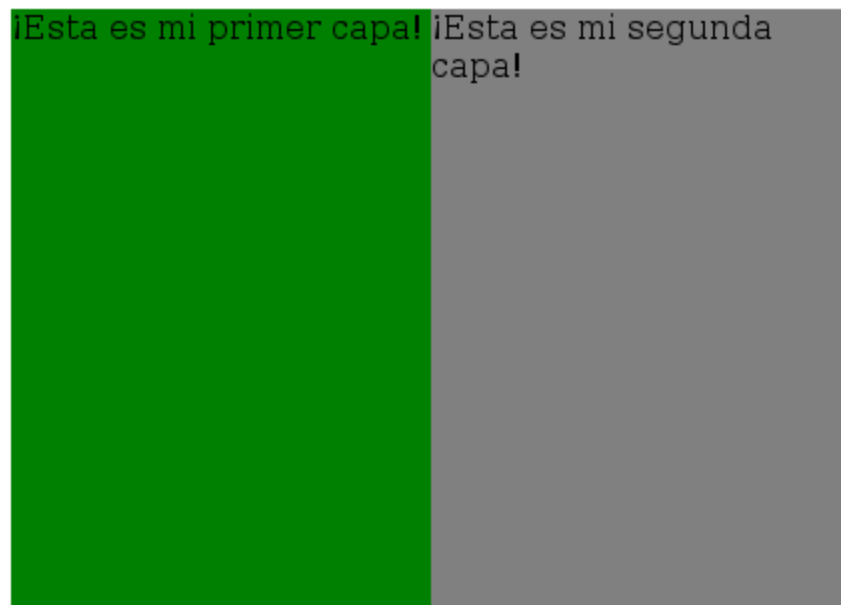
```
float:left;  
}
```

y el código HTML sería este:

```
<html>  
  <head>  
    <title>Curso de maquetacion CSS</title>  
    <link href="estilos.css" type="text/css" rel="stylesheet">  
  </head>  
  <body>  
    <div id="capa1">¡Esta es mi primer capa!</div>  
    <div id="capa2">¡Esta es mi segunda capa!</div>  
  </body>  
</html>
```

**Nota:** Asegurarse de escribir bien los ID, tanto en la hoja de estilos como en el código HTML, que coincidan en ambos, de lo contrario no funcionará.

Esto se debería ver más o menos así:



Ahora, ¿que pasa si aplicamos un margen izquierdo a la capa2?

```
#capa2{  
  width:210px;  
  height:300px;  
  background-color:gray;  
  float:left;  
  margin-left:10px;  
}
```

se vería así:



#### 4. Evitar que un DIV o capa se superponga con otra

Seguramente cuando estés maquetando tu sitio web necesitarás tener una capa o bloque que no tenga capas a su/s lados, para eso esta la propiedad **CSS Clear**.

Esta propiedad se utiliza en conjunto con *float* y sirve para evitar que una capa se posicione a cualquiera de los lados, los valores que admite son estos:

- **left**: no deja que una capa flote a la izquierda
- **right**: evita que una capa flote a la derecha
- **both**: evita que haya capas flotantes en cualquiera de sus lados

Nosotros lo usaremos para crear, por ejemplo, el pie de página.

Siguiendo con el ejemplo que hicimos en la lección anterior vamos a crear una tercer capa #capa3 y le aplicaremos esta propiedad, añadiremos un ancho de 430 pixeles y un alto de 30 pixeles, también le aplicaremos un color de fondo naranja.

```
#capa3{  
    width:430px;  
    height:30px;  
    background-color:orange;  
    float:left;  
    clear:both;  
}
```

el código HTML quedaría así:

```

<html>
  <head>
    <title>Curso de maquetacion CSS</title>
    <link href="estilos.css" type="text/css" rel="stylesheet">
  </head>
  <body>
    <div id="capa1">¡Esta es mi primer capa!</div>
    <div id="capa2">¡Esta es mi segunda capa!</div>
    <div id="capa3">¡Esta es mi tercer capa!</div>
  </body>
</html>

```

Esto dará como resultado algo parecido a esto:



Ahora vamos a agregar un margen superior para separar un poco el pie de página (#capa3) de las demás capas o divs.

```

#capa3{
  width:430px;
  height:30px;
  background-color:orange;
  float:left;
  clear:both;
  margin-top:10px;
}

```

Esto quedaría así:



Fin.