

## 02 Posicionamiento de cajas

Ya una vez que tenemos los principales aspectos de las cajas, vamos a revisar ahora como posicionarlas en nuestra página, para iniciar vamos a crear varios elementos div, con la clase box y al segundo de ellos le vamos a poner una clase llamada position, que es con la cual vamos a entender el posicionamiento.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <link rel="stylesheet" href="styles/styles.css">
7    <title>Posicionamiento de cajas</title>
8  </head>
9  <body>
10   <div class="box"></div>
11   <div class="box position"></div>
12   <div class="box"></div>
13   <div class="box"></div>
14   <div class="box"></div>
15   <div class="box"></div>
16   <div class="box"></div>
17   <div class="box"></div>
18   <div class="box"></div>
19   <div class="box"></div>
20 </body>
21 </html>
```

A los estilos vamos a aplicarle los siguientes cambios:

```
1  * {
2    margin: 0;
3    padding: 0;
4  }
5
6  body {
7    background: #f2f2f2;
8  }
```

Nótese que en la línea 1, en vez de colocar el nombre de una etiqueta o una clase, estamos usando un comodín (\*). Con esto estamos indicando que a todos los componentes les vamos a poner inicialmente un atributo de margen y padding en 0px, con ello aseguramos que todos los elementos se ubiquen con una margen de cero tanto con la parte superior del navegador

como con los bordes laterales<sup>1</sup>. Note también que en vez de colocar 0px, pusimos simplemente 0 (cero) esto lo podemos hacer sólo cuando el valor es cero, en cualquier otro caso debemos especificar la unidad.

Hasta el momento visualmente no tenemos ningún cambio, vamos entonces a agregar la clase box con los siguientes atributos:

```
.box{
  width:200px;
  height:200px;
  background: #4e73df;
  margin: 20px;
}
```

Para entender las diferentes formas del posicionamiento, vamos a tomar la segunda caja, a la cual le pusimos la clase position y a modificar algunos atributos. Primero modificaremos su color de fondo, para distinguirla fácilmente de las otras, seguido de ello modificamos la propiedad **position**, que es precisamente la que nos permite posicionar el elemento. El valor por defecto de esta propiedad es **static**, por ello si hacemos el cambio:

```
.position{
  background: #086788;
  position: static;
}
```

El único cambio visualmente será el color de fondo de la caja:



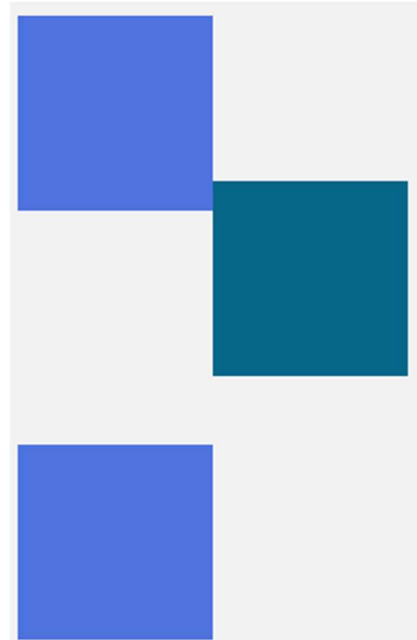
---

<sup>1</sup> Esto es una práctica muy común debido a que por defecto el navegador suele colocar márgenes y a veces necesitamos que nuestros componentes queden con un ancho completo y sin espacios en la parte superior.

Vamos a cambiar la propiedad **position** con el valor **relative**, el efecto es que la caja se puede desplazar una distancia relativa a su posición original. Para desplazar la caja, necesitamos utilizar las propiedades **top**, **right**, **left** y **bottom**, con ellas indicamos cuantas unidades queremos desplazar el elemento.

```
.position{  
  background: #086788;  
  position: relative;  
  left: 200px;  
  bottom: 50px;  
}
```

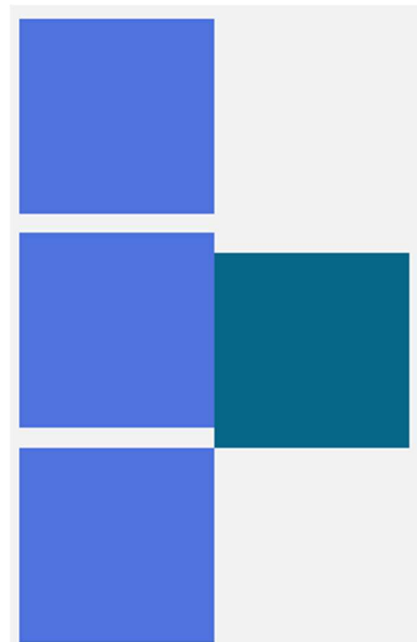
En este caso desplazamos 200px hacia la izquierda y 50px desde el fondo hacia arriba medidos desde su posición original. EN las medidas de desplazamiento, se pueden utilizar valores negativos, el efecto visual será el desplazamiento en el sentido contrario, es decir, si colocamos **bottom: -50px**; en vez de mover de abajo hacia arriba 50px, el desplazamiento será desde arriba hacia abajo.



El siguiente tipo de posicionamiento es el **absoluto** (absolute).

```
.position{  
  background: #086788;  
  position: absolute;  
  left: 200px;  
}
```

Con el posicionamiento absoluto, también hay un desplazamiento de la caja, pero en este caso, los otros elementos se desplazan ocupando su lugar. La diferencia principal entre relativo y absoluto es que, en el posicionamiento relativo, el componente se desplaza, pero su ubicación original se conserva en blanco, en el caso del absoluto, este desplazamiento hace que la ubicación original del componente sea sustituida por el siguiente componente.



Si en vez de colocar 200px, ponemos **left: 150px;** tendríamos lo siguiente:



En esta ocasión la cantidad de px desplazados es menor al ancho del siguiente componente, el efecto de esto es una sobreposición de los elementos.

Otro tipo de posicionamiento es el fijo (fixed). Visualmente es similar al absoluto, pero la gran diferencia es que el elemento se queda fijo en la pantalla, incluso al tratar de mover la pantalla de arriba hacia abajo o a los lados, el componente seguirá en el mismo, mientras los otros componentes se mueven a su alrededor.

```
.position{  
  background: #086788;  
  position: fixed;  
  left: 150px;  
}
```



Esto es muy útil cuando queremos tener, por ejemplo, un menú en la parte superior y que permanezca fijo todo el tiempo, así nos estemos desplazando hacia debajo de la página.

Antes de ver el último tipo de posicionamiento, vamos a hacer algunas modificaciones

Tomamos el componente con el que hemos venido probando y lo colocamos dentro de otro elemento de tipo div. Este nuevo div tendrá la clase **parent-box**

```
<div class="box"></div>  
<div class="parent-box">  
  <div class="box position"></div>  
</div>  
<div class="box"></div>  
<div class="box"></div>
```

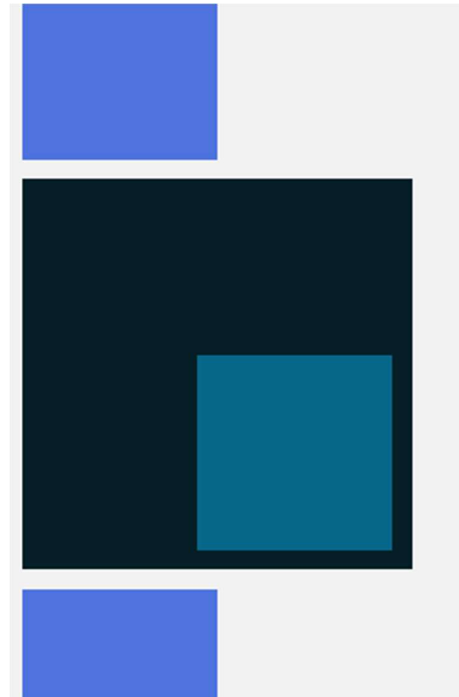
A la nueva clase (parent-box) le damos un tamaño más grande, una posición relativa y un color más oscuro para poder destacarla de los otros elementos.

```
.parent-box{  
  margin: 20px;  
  background: #061f27;  
  position: relative;  
  width:400px;  
  height:400px;  
}
```

A la clase position, le modificamos su atributo position, bottom y right, observe que para posicionar también podemos usar porcentajes.

```
.position{  
  background: #086788;  
  position: absolute;  
  bottom: 0%;  
  right: 0%;  
}
```

El resultado final es que tenemos una caja padre, que contiene una caja hija que se posiciona con relación a su padre. La clave de lo que acabamos de hacer es que la caja padre tenga el posicionamiento relativo y el hijo, absoluto. Trata de combinar estas dos posiciones con el padre y el hijo y observa qué otros resultados tienes.



Para el último ejercicio, vamos a devolver los cambios en el *html* que hicimos anteriormente para dejar nuestras cajas como estaban al principio. En el *css*, vamos a modificar nuestra clase *position* y a dejarle únicamente el *background*, finalmente, a nuestra clase *box*, vamos a agregarle el atributo ***float: left;***

```
.box{  
  width:200px;  
  height:200px;  
  background: #4e73df;  
  margin: 20px;  
  float: left;  
}  
  
.position{ background: #086788; }
```

Deberíamos observar el siguiente resultado:



La propiedad ***float*** lo que hace en este caso es que organiza los elementos de derecha a izquierda (*float:right;* ordenaría de izquierda a derecha). Esto también es una manera de que un elemento de tipo bloque se posicione al lado de otro.