

Position

position:sticky

Actúa como si estuviera posicionado con el valor relative, hasta que se alcanza un umbral de desplazamiento cambiando su valor a fixed.

Dicho umbral es ajustado con las propiedades top, bottom, left, right.

Por ejemplo: `top:0px; /*Cuando llegue a la posición 0px pasa a fixed*/`

Ejercicio

Utilice su ejercicio anterior de Menu bar.

Agregue un div de 300px de alto y 100% de ancho antes y después del menú bar.

Cambie el `position:fixed` de menú bar a `position:sticky`. Por último indique el punto de cambio a `fixed`.

Ejemplo: `top: 0px;`

Ahora desplace el scroll vertical para notar el efecto.



Solución

```
body { height: 1500px; }

.box {
  height: 300px;
  width: 100%;
}

.color-coral { background-color: coral; }
.color-green { background-color: rgb(25, 214, 0); }

.menu {
  position: sticky;
  top: 0px;
  width: 100%;
}

.menu > ul {
  list-style: none;
  overflow: hidden;
  padding: 0px;
  margin: 0px;
  background-color: teal;
}

.menu li {
  float: left;
  border-right: 1px solid rgb(0, 80, 80);
}
```

```
.menu a {
  display: block;
  padding: 10px 10px 10px 10px;
  text-decoration: none;
  color: white;
  transition: background-color 0.5s;
}
```

```
<div class="box color-coral"></div>
<nav class="menu">
  <ul>
    <li><a href="#">Inicio</a></li>
    <li><a href="#">Noticias</a></li>
    <li><a href="#">Entretenimiento</a></li>
    <li><a href="#">Contacto</a></li>
    <li><a href="#">Soporte</a></li>
  </ul>
</nav>
<div class="box color-green"></div>
```

position:absolute

El elemento es removido del flujo normal del documento. Es posicionado de forma relativa a su ancestro más cercano (que tenga cualquier valor de position excepto static) , si no lo hay, se referencia a la ventana del navegador. Su posición final está determinada por los valores de top, right, bottom, y left.

Escriba el siguiente código

```
.main {  
  position: relative;  
  
  width: 500px;  
  height: 500px;  
  background-color: coral;  
}  
  
.box-normal {  
  width: 250px;  
  height: 250px;  
  background-color: crimson;  
}  
  
.box-absolute {  
  position: absolute;  
  
  width: 100px;  
  height: 100px;  
  background-color: teal;  
}  
  
.box-relative {  
  position: relative;  
  
  width: 25px;  
  height: 25px;  
  background-color: cyan;  
}
```

```
<div class="main">  
  <div class="box-normal">  
    <div class="box-absolute"></div>  
    <div class="box-relative"></div>  
  </div>  
</div>
```



Pregunta 1

¿Por qué la caja box-relative está en el interior de box-absolute? ¿Por qué no se encuentra debajo de ella?

box-relative



box-absolute



```
<div class="main">  
  <div class="box-normal">  
    <div class="box-absolute"></div>  
    <div class="box-relative"></div>  
  </div>  
</div>
```

Respuesta

Porque la caja box-absolute tiene un **position: absolute**, esto quiere decir que está fuera del flujo, por lo tanto box-relative no sabe que se encuentra ahí y se posiciona en su lugar.

Pregunta 2

¿Qué modificación debo hacer para que box-relative vea a box-absolute?

Respuesta 2

Box-absolute deberá cambiar su position a static (Valor por defecto) o a relative.

```
.box-absolute {  
  position: relative;  
  
  width: 100px;  
  height: 100px;  
  background-color: teal;  
}
```



Pregunta 3 (Usando el código inicial)

¿Qué sucede si muevo box-normal a otra posición con top/left?

Respuesta 3

Nada, top y left no funcionan.

Para funcionar se requiere que position tenga un valor diferente a static.

Pregunta 4 (Usando el código inicial)

¿Qué sucede si agrego un margin-left a box-normal?

¿Qué sucede con los div interno de box-normal?

Respuesta 4

Se desplaza la cantidad indicada en pixeles hacia la derecha.


Los hijos internos se desplazan junto con él.

```
.box-normal {  
  width: 250px;  
  height: 250px;  
  background-color: crimson;  
  margin-left: 100px;  
}
```



Pregunta 5 (Usando el código inicial)

Aplicando la siguiente propiedad a box-normal

```
.box-normal {  
    width: 250px;  
    height: 250px;  
    background-color:  crimson;  
    margin-left: 100px;  
}
```

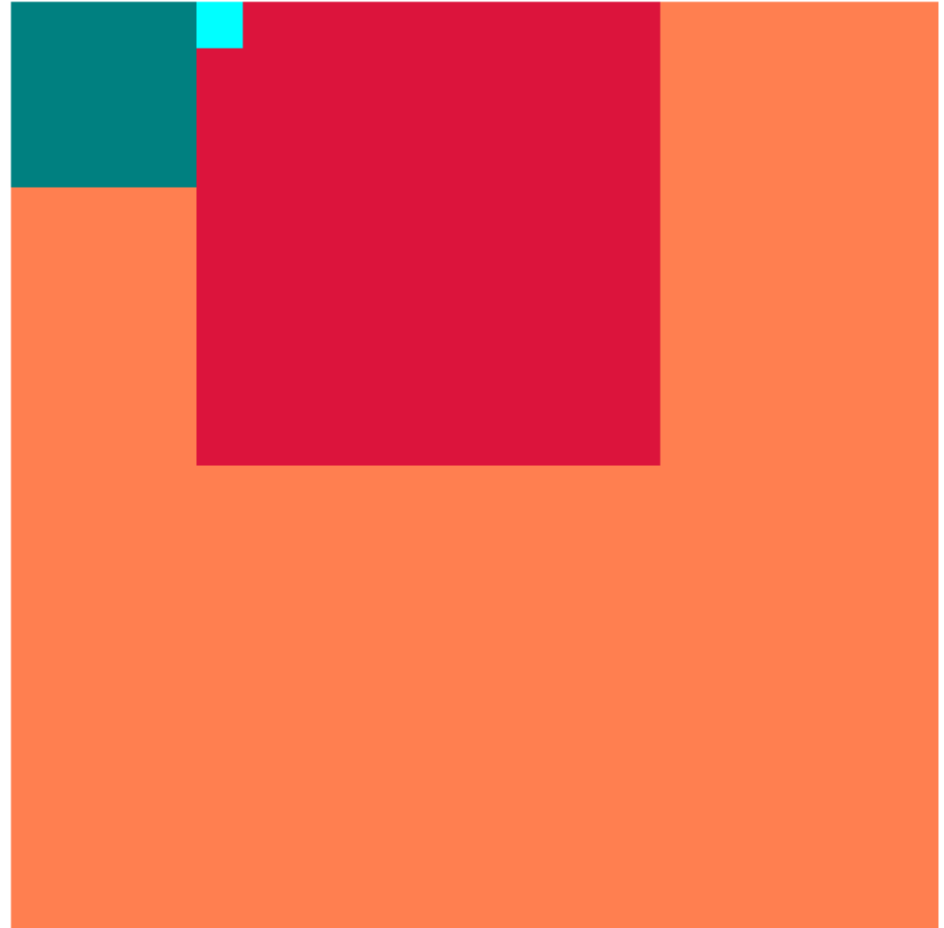
¿Qué sucede si agrego un `left: 0px;` a `box-absolute`?

Respuesta

Se posiciona en el borde de su contenedor padre **referencial**.

Su contenedor de referencia será el más próximo que contenga un position diferente a static.

Por lo tanto, su contenedor de referencia es **box-main**.



Pregunta 6 (Usando el código de la **pregunta anterior**)

¿Qué cambio debo realizar para que al aplicar `left:0px;` en `box-absolute` se vea así:



Y no así:




```
.box-normal {  
  width: 250px;  
  height: 250px;  
  background-color: crimson;  
  
  margin-left: 100px;  
}  
  
.box-absolute {  
  position: absolute;  
  width: 100px;  
  height: 100px;  
  background-color: teal;  
  
  left: 0px;  
}
```

Respuesta 6

box-absolute tomará de referencia el contenedor más cercano que contenga un position diferente a static.

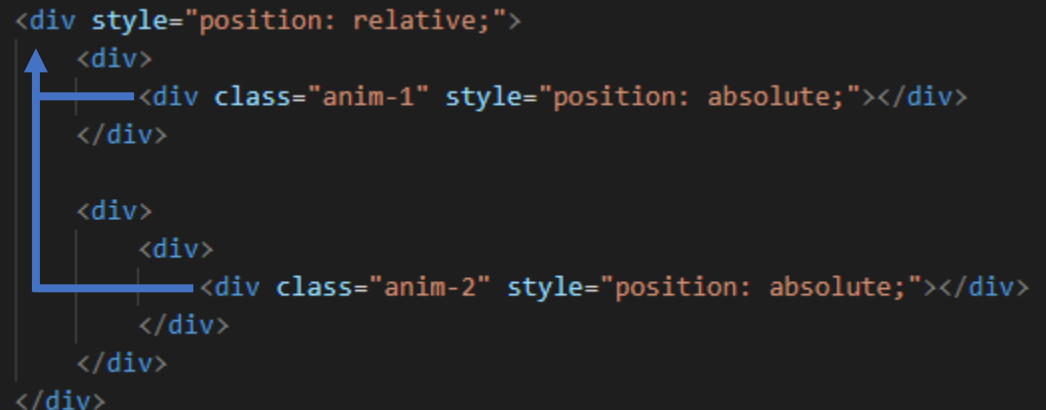
Por lo tanto, habría que agregar cualquier tipo de position (menos static) a box-normal.

```
.box-normal {  
  position: relative;  
  
  width: 250px;  
  height: 250px;  
  background-color:  crimson;  
  
  margin-left: 100px;  
}
```

Observación

Aunque absolute puede parecer un poco más complejo que relative, es bastante útil, sobre todo para realizar animaciones en la que se necesita que no afecte a otros elementos y todas las animaciones deban usar **un mismo punto** de referencia, sin importar el nivel de anidación que tengan los elementos.

Al ocupar top/bottom/left/right ocupan de referencia el primer contenedor.





The diagram illustrates a nested HTML structure where absolute positioning is used relative to a common ancestor. A blue L-shaped arrow points from the top-left corner of the outermost container to the top-left corner of the innermost container, indicating that the absolute positions are calculated relative to the same reference point (the top-left of the outermost container).

```
<div style="position: relative;">  
  <div>  
    <div class="anim-1" style="position: absolute;"></div>  
  </div>  
  <div>  
    <div>  
      <div class="anim-2" style="position: absolute;"></div>  
    </div>  
  </div>  
</div>
```

Ejercicio

Escriba el siguiente código.

Pruebe cada propiedad (top/bottom/left/right) por separado. Asigne el valor de 0px y observe el comportamiento.

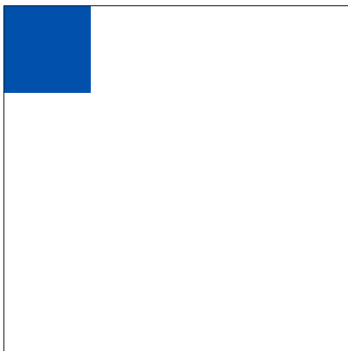
```
.main {  
  position: relative;  
  height: 400px;  
  width: 400px;  
  border: 1px solid  black;  
}  
  
.box {  
  position: absolute;  
  width: 100px;  
  height: 100px;  
  background-color:  rgb(0, 80, 172);  
}
```

```
<div class="main">  
  <div class="box"></div>  
</div>
```

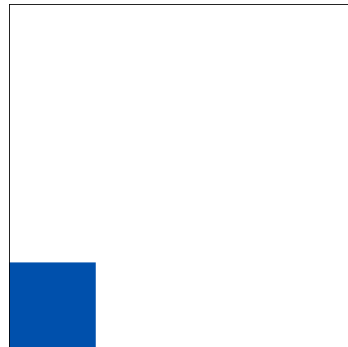
Resultado

Al usar cada propiedad por separado ingresando 0px, el cuadrado azul se posicionará en el punto inicial de cada borde referente a su contenedor padre.

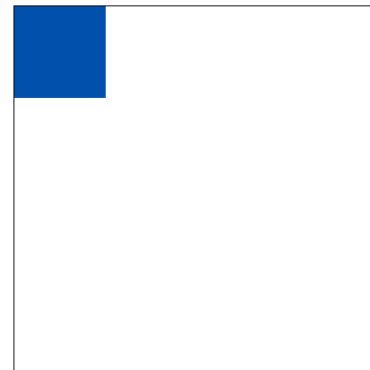
top :0px



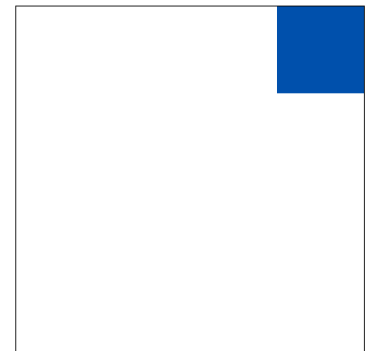
bottom:0px



left:0px





right:0px




Ejercicio

Escriba los siguientes códigos y observe su comportamiento.

```
.box {  
  position: absolute;  
  width: 200px;  
  background-color: rgb(0, 80, 172);  
  
  top: 50px;  
  bottom: 0px;  
}
```

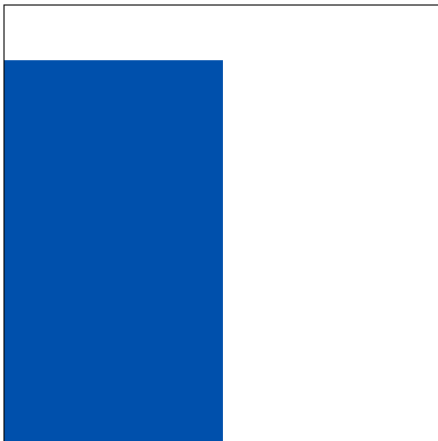
```
.box {  
  position: absolute;  
  height: 200px;  
  background-color: rgb(0, 80, 172);  
  
  right: 0px;  
  left: 0px;  
}
```

```
.box {  
  position: absolute;  
  
  background-color: rgb(0, 80, 172);  
  
  top: 50px;  
  right: 0px;  
  left: 0px;  
  bottom: 50px;  
}
```

Resultado

Cuando no se indica el ancho o alto y se utiliza las propiedades top/bottom/right/left estas deformarán la caja respecto a los valores indicados.

**top: 50px;
bottom: 0px;**



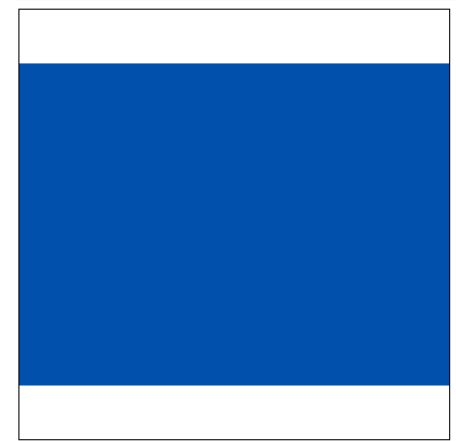
widht:200px

**right: 0px;
left: 0px;**



height:200px

**top: 50px;
bottom: 50px
right: 0px;
left: 0px;**



Sin width/height

Recordatorio - overflow

La propiedad **overflow:hidden;** oculta todo el contenido que se desborde del contenedor que lleve esta propiedad.

Otras variantes.

overflow-x/ overflow-y: Controlan el des-border hacia una dirección.

Ejercicio

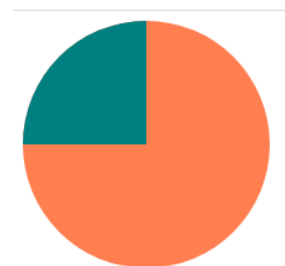
Implemente el siguiente código

```
.circle {  
  height: 200px;  
  width: 200px;  
  border-radius: 200px;  
  background-color: coral;  
}  
  
.box {  
  width: 100px;  
  height: 100px;  
  background-color: teal;  
}
```

```
<div class="circle">  
  <div class="box"></div>  
</div>
```



¿Qué se debe hacer para que solamente se vea la parte del cuadro que está en el interior del círculo?



Respuesta

A la clase `.circle` se le debe agregar la propiedad `overflow:hidden`;

```
.circle {  
  height: 200px;  
  width: 200px;  
  border-radius: 200px;  
  background-color: coral;  
  overflow: hidden;  
}
```

Propiedad - opacity

En los colores la transparencia se maneja con el canal Alpha, y en los elementos(etiquetas) es posible manejar la transparencia con la propiedad opacity.

Si a un contenedor se le agrega la propiedad opacity y contiene hijos, ellos también serán afectados.

Acepta un número entero o decimal entre 0 y 1.

Ejercicio

Agregue una imagen que por defecto su opacidad sea cero y al momento de acerca el mouse debe aparecer con una transición.



Solución

```
.container > img {  
  opacity: 0;  
  transition: opacity 0.5s;  
}  
  
.container > img:hover {  
  opacity: 1;  
}
```

```
<div class="container">  
    
</div>
```