

Position

Position

La propiedad position de CSS, especifica cómo un elemento es posicionado en el documento. Las propiedades top, right, bottom, y left determinan la ubicación final de los elementos.

Podemos indicar si un elemento debe subir o bajar, desplazarse a la izquierda o derecha (Efecto parecido a margin con la diferencia que no afecta al resto de elementos).

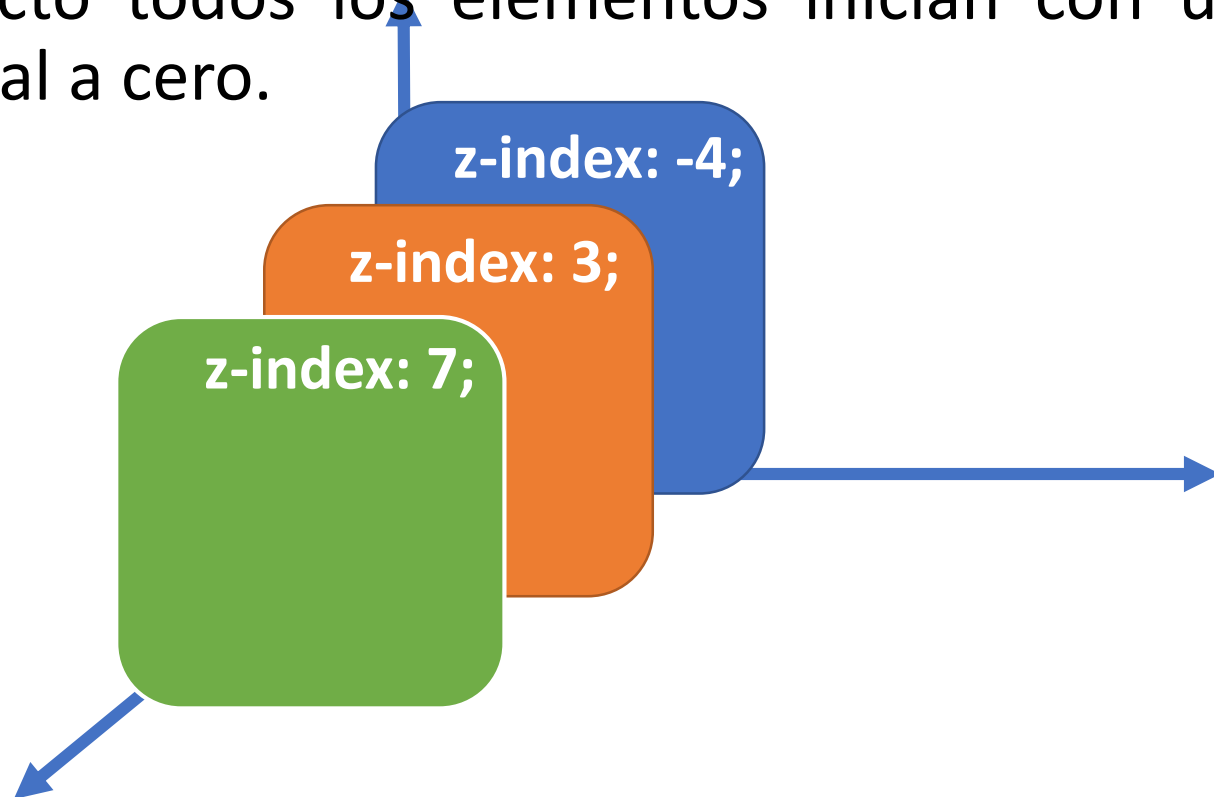
position: static

El elemento es posicionado de acuerdo con el flujo normal del documento. Las propiedades top, right, bottom, left, and z-index no tienen efecto. Este es el valor por defecto para cualquier elemento.

Recordatorio - z-index

Cuando varios elementos se superponen, z-index determina cuales van sobre otros (Imaginar una pila de hojas). Por lo general un elemento con mayor z-index se posicionará por delante de uno menor.

Por defecto todos los elementos inician con un z-index igual a cero.



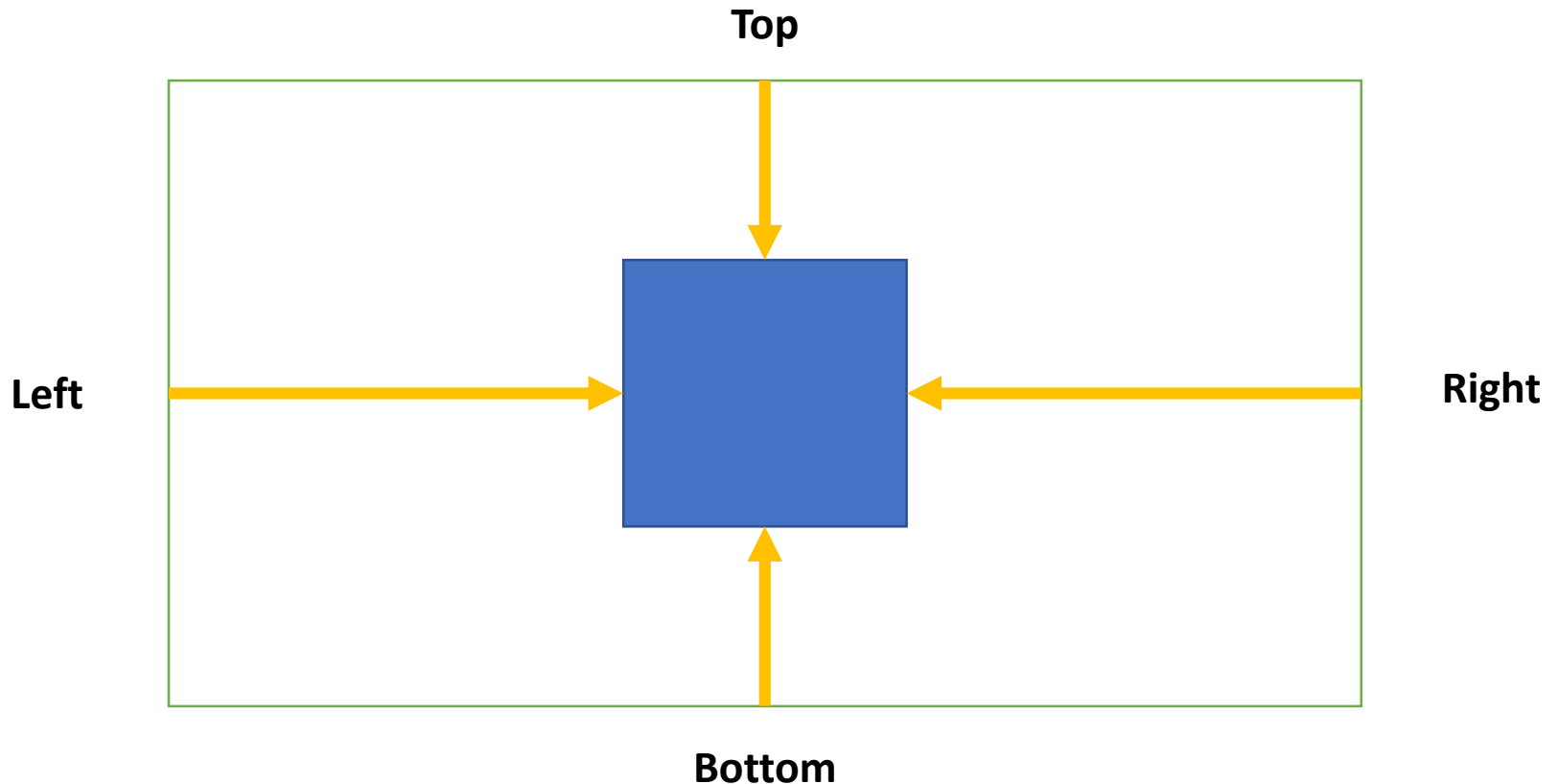
position:relative

El elemento es posicionado de acuerdo con el flujo normal del documento, y luego es desplazado con relación a sí mismo, con base en los valores de top, right, bottom, y left. El desplazamiento **no afecta la posición de ningún otro elemento**; por lo que, el espacio que se le da al elemento en el esquema de la página es el mismo como si la posición fuera static. Este valor crea un nuevo contexto de apilamiento, donde el valor de **z-index** no es auto.

Observación

Las propiedades top/bottom/left/right toman de referencia el punto de referencia inicial o dependiendo del position a su contenedor padre.

Observe las flechas amarillas, un valor positivo para cualquiera de esas propiedades moverá la caja en esa dirección.



Recomendación

Es recomendable trabajar solamente bajo 2 ejes en el caso que se quiera mover un elemento.

Los valores asignados pueden ser positivos como negativos.

Elegir entre arriba o abajo e izquierda o derecha.

1. top/left
2. top/rigth
3. bottom/left
4. bottom/right

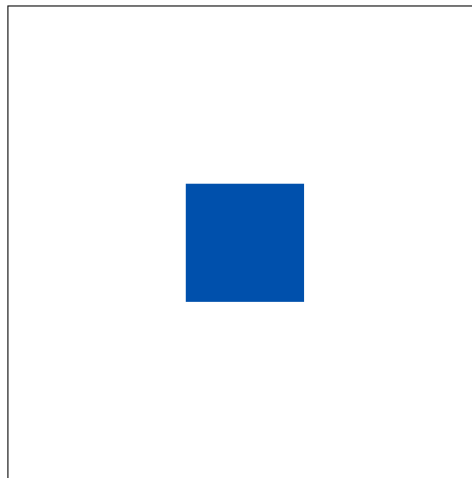
Ejercicio

Intente dejar la caja azul lo más centrada posible (No es necesario exactitud).

Solucione este problema 4 veces. Utilice una configuración diferente en cada solución.

1. Solucionar con: **top/left**
2. Solucionar con: **top/right**
3. Solucionar con: **bottom/left**
4. Solucionar con: **bottom/right**

Consejo: Al inicio aplique valores pequeños para detectar hacia qué lado se mueve la caja.



```
.main {
  height: 400px;
  width: 400px;
  border: 1px solid black;
}

.box {
  position: relative;
  width: 100px;
  height: 100px;
  background-color: rgb(0, 80, 172);
  /*
    left: ??px;
    right: ??px;
    top: ??px;
    bottom: ??px;
  */
}
```

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


Soluciones

top/left

```
.box {  
  position: relative;  
  width: 100px;  
  height: 100px;  
  background-color: ■rgb(0, 80, 172);  
  
  top: 150px;  
  left: 150px;  
}
```

top/right

```
.box {  
  position: relative;  
  width: 100px;  
  height: 100px;  
  background-color: ■rgb(0, 80, 172);  
  
  top: 150px;  
  right: -150px;  
}
```

bottom/left

```
.box {  
  position: relative;  
  width: 100px;  
  height: 100px;  
  background-color: ■rgb(0, 80, 172);  
  
  bottom: -150px;  
  left: 150px;  
}
```

bottom/right

```
.box {  
  position: relative;  
  width: 100px;  
  height: 100px;  
  background-color: ■rgb(0, 80, 172);  
  
  bottom: -150px;  
  right: -150px;  
}
```

Otra forma para centrar


Es posible calcular el punto central bajo la siguiente ecuación:

$$y = \text{Altura Contenedor} / 2 - \text{Altura Hijo} / 2$$

$$x = \text{Ancho Contenedor} / 2 - \text{Ancho Hijo} / 2$$

En css existe la función `calc()` para poder realizar una operación aritmética.

Utilizando el ejercicio anterior - div hijo de 100x100.

```
.box {  
  position: relative;  
  width: 100px;  
  height: 100px;  
  background-color:  rgb(0, 80, 172);  
  
  top: calc(100% / 2 - 100px / 2); /* calc(50% - 50px); */  
  left: calc(100% / 2 - 100px / 2); /* calc(50% - 50px); */  
}
```

(El signo de la operación
debe ir separado del
segundo valor)

Pregunta

¿Qué sucederá si aplico top y bottom al mismo tiempo?

¿Qué sucederá si aplico left y right al mismo tiempo?

Respuesta

En el caso de top/bottom, solamente se toma en cuenta la propiedad top. (Bottom es ignorado)

En el caso de left/right, solamente se toma en cuenta la propiedad left. (Right es ignorado)

Ejercicio

Escriba el siguiente código y observe su comportamiento.

¿Qué sucede si los valores de top/left cambian? ¿Afectará al otro Div?

```
<section class="box">
  <div class="div-1"></div>
  <div class="div-2"></div>
</section>
```

```
section {
  width: 300px;
  height: 200px;
  border: 1px solid black;
}

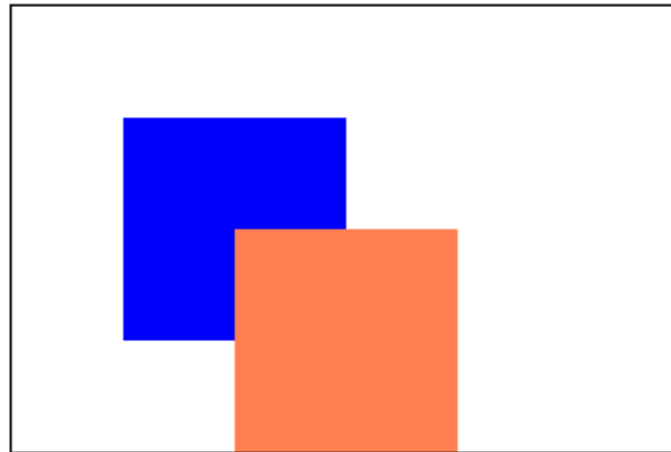
div {
  height: 100px;
  width: 100px;
  position: relative;
}

.div-1 {
  background-color: blue;
  top: 50px;
  left: 50px;
}

.div-2 {
  background-color: coral;
  top: 0px;
  left: 100px;
}
```

Respuesta

Como el position de los divs es relativo, no afectará al flujo de los demás elementos.



Agregue un position: relative al contenedor padre y asígnele un top: 50px y left: 50px. ¿Los elementos hijos se moverán con el?

Ejercicio

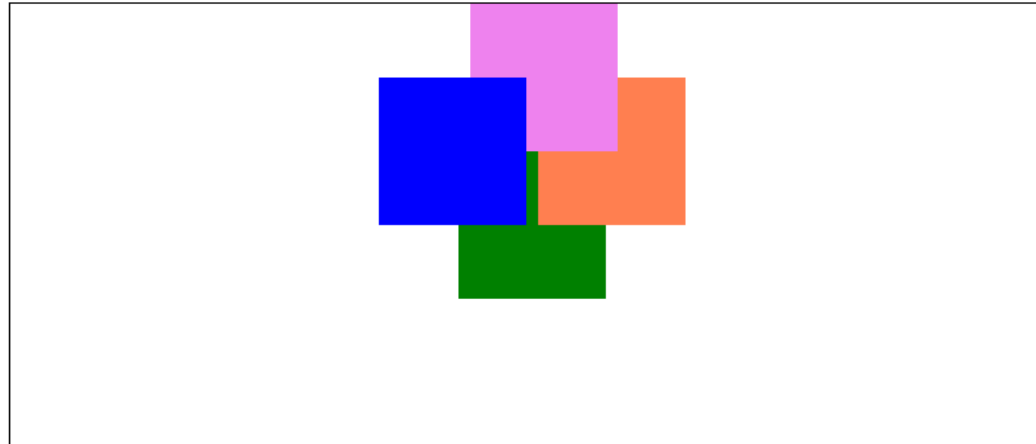
Implemente el código necesario para representar la siguiente imagen.

Observación: Utilice z-index.




El contenedor tiene una dimensión de 700x300.



Para los 4 divs de colores:

1. Todos tienen un `display:inline-block`;
2. Los **3** divs iniciales tiene `position:relative`;
3. Todos tiene una dimensión de 100x100



Solución

```
section {  
  border: 1px solid  black;  
  height: 300px;  
  width: 700px;  
}  
  
div {  
  height: 100px;  
  width: 100px;  
  display: inline-block;  
}  
  
.relative { position: relative; }  
  
.blue {  
  background-color:  blue;  
  top: 50px;  
  left: 250px;  
}  
  
.green {  
  background-color:  green;  
  top: 100px;  
  left: 200px;  
  z-index: -2;  
}
```

```
.coral {  
  background-color:  coral;  
  top: 50px;  
  left: 150px;  
  z-index: -1;  
}  
  
.violet { background-color:  violet; }
```

```
<section>  
  <div class="blue relative"></div>  
  <div class="green relative"></div>  
  <div class="coral relative"></div>  
  <div class="violet"></div>  
</section>
```


position: fixed

El elemento es removido del flujo normal del documento.

Los elementos con position: fixed toman como referencia la ventana del navegador y no respetan el tener un contenedor padre que esté posicionado. Además, al hacer scroll en la página, el elemento que esté posicionado como position: fixed seguirá en la misma posición respecto a la ventana del navegador, aunque el scroll haya desplazado la página hacia abajo. Útil para menús siempre visibles.

Ejercicio

Implemente el siguiente menú en la parte superior de la página.

Por último agregue al `<body>` un `height` de 1500px o más, con el fin de hacer aparece el scroll vertical.

Recuerde agregar `position:fixed;` al contenedor de menú.



Solución

```
body { height: 1500px; }

.menu {
  position: fixed;
  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;
}
```

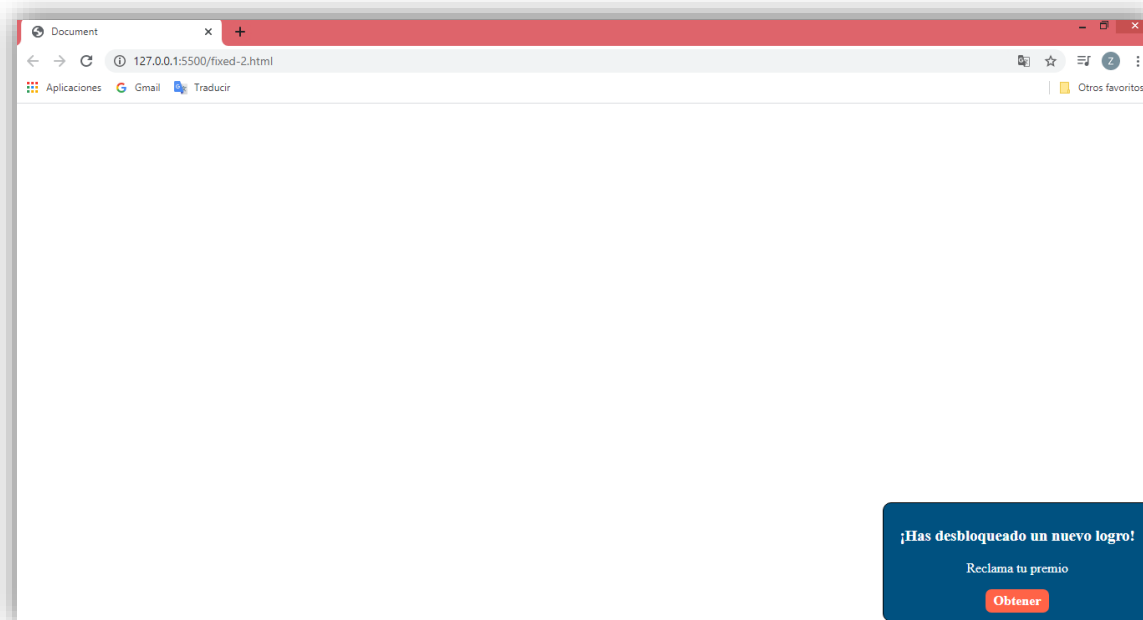
Ejercicio

Diseña un Panel de ayuda que se encuentre siempre en el lado derecho inferior de la pantalla.




¡Has desbloqueado un nuevo logro!


Reclama tu premio

Obtener



Solución

```
.message{  
  position:fixed;  
  bottom:5px;  
  right: 5px;  
  
  width: 300px;  
  background-color: rgb(0, 81, 128);  
  color:white;  
  padding: 10px;  
  text-align: center;  
  border: 1px solid black;  
  border-radius: 10px;  
}
```

```
.message-reward{  
  background-color: tomato;  
  font-weight: bolder;  
  border-radius: 7px;  
  padding: 5px 10px 5px 10px;  
  /*fit-content - Revisar Prefijo para otros navegadores*/  
  width: fit-content;  
  margin: 0 auto;  
}
```

```
<div class="message">  
  <h3>¡Has desbloqueado un nuevo logro!</h3>  
  <p>Reclama tu premio</p>  
  <div class="message-reward">Obtener</div>  
</div>
```