

# 35 CSS: Posicionamiento **relative**

El posicionamiento relativo ( `position: relative;` ) es el primer paso para salir del flujo completamente estático sin causar grandes alteraciones en el layout general. Permite ajustar la posición de un elemento respecto a dónde se habría colocado *normalmente*.

## Características Principales:

1. **Permanece en el Flujo Normal (Inicialmente):** Un elemento con `position: relative;` se coloca inicialmente en la página como si fuera `static`, siguiendo el flujo normal del documento.
2. **Desplazamiento con `top`, `right`, `bottom`, `left`:** A diferencia de `static`, las propiedades de desplazamiento ( `top`, `right`, `bottom`, `left` ) **sí funcionan** con `position: relative;`. Estos valores desplazan al elemento **desde su posición original** en el flujo.
  - `top: 10px;` : Mueve el elemento 10px hacia abajo desde su posición superior original.
  - `left: 20px;` : Mueve el elemento 20px hacia la derecha desde su posición izquierda original.
  - `bottom: 10px;` : Mueve el elemento 10px hacia arriba desde su posición inferior original.
  - `right: 20px;` : Mueve el elemento 20px hacia la izquierda desde su posición derecha original.
  - Se pueden usar valores positivos y negativos.
3. **Reserva de Espacio Original:** Este es un punto clave: aunque el elemento se desplace visualmente, el espacio que *originalmente* ocupaba en el flujo normal **sigue reservado**. Otros elementos no se moverán para ocupar ese espacio vacío. Esto evita que el layout circundante se "rompa" o reajuste drásticamente cuando aplicas un pequeño desplazamiento.
4. **Crea Contexto de Posicionamiento para Hijos Absolutos:** Un elemento con `position: relative;` (y también `absolute`, `fixed`, `sticky`) crea un **nuevo contexto de posicionamiento** para sus elementos descendientes que tengan `position: absolute;`. Esto significa que un hijo con `position: absolute;` se posicionará con respecto a los bordes de su ancestro relativo más cercano, en lugar de hacerlo respecto al `<html>` o la ventana. Este es uno de los usos más importantes de `position: relative;`.
5. **Funciona con `z-index`:** Los elementos con `position: relative;` pueden usar la propiedad `z-index` para controlar su orden de apilamiento si se superponen con otros elementos posicionados.

## Ejemplo Visual:

### • HTML:

```
<div class="contenedor">
  <div class="caja static">Caja Estática 1</div>
  <div class="caja relativa">Caja Relativa (Desplazada)</div>
  <div class="caja static">Caja Estática 2</div>
</div>
```

### • CSS:

```
.contenedor {
  border: 1px dashed grey;
  padding: 10px;
```

```

margin-top: 30px;
}
.caja {
width: 150px;
height: 80px;
padding: 10px;
margin-bottom: 10px; /* Espacio normal entre cajas */
color: white;
text-align: center;
}
.static {
background-color: steelblue;
}
.relativa {
position: relative; /* Activamos posicionamiento relativo */
background-color: tomato;
/* Desplazamos la caja */
top: 20px; /* 20px hacia abajo desde su posición original */
left: 30px; /* 30px hacia la derecha desde su posición original */
z-index: 1; /* Aseguramos que esté por encima si se solapa */
}

```

#### • Resultado:

1. La "Caja Estática 1" se mostrará normalmente.
2. La "Caja Relativa" se dibujará visualmente 20px más abajo y 30px más a la derecha de donde *debería* haber estado (justo debajo de la Caja 1, con 10px de margen).
3. Se verá un **espacio vacío** donde la "Caja Relativa" *estaría* si no se hubiera desplazado.
4. La "Caja Estática 2" aparecerá debajo de ese espacio vacío, *no* directamente debajo de la "Caja Estática 1", porque el espacio de la caja relativa sigue reservado en el flujo.
5. La "Caja Relativa" podría superponerse visualmente con la "Caja Estática 2" debido a su desplazamiento.

#### Casos de Uso Comunes para `position: relative;`:

1. **Pequeños Ajustes Visuales:** Mover ligeramente un elemento de su posición normal sin afectar al resto del layout.
2. **Crear un Contexto de Posicionamiento:** Establecer `position: relative;` en un contenedor (a menudo sin `top/left/bottom/right`, dejándolo en `0`) para que sirva de referencia a elementos hijos con `position: absolute;`. Este es, quizás, su uso más frecuente y fundamental en diseños complejos.
3. **Controlar el `z-index`:** Aplicar `position: relative;` a un elemento (incluso sin desplazarlo) para poder asignarle un `z-index` y controlar su apilamiento respecto a otros elementos posicionados.

#### Conclusión:

`position: relative;` ofrece una forma de ajustar la posición de un elemento respecto a su ubicación normal en el flujo, sin eliminarlo de él y conservando su espacio original. Su capacidad para crear contextos de posicionamiento para elementos absolutos lo convierte en una herramienta

indispensable para construir layouts complejos y controlar la disposición de elementos de forma precisa.

---

Ahora que hemos visto `relative`, el siguiente paso lógico es entender `position: absolute`, ya que a menudo trabajan juntos. ¿Continuamos?