

34 CSS: Posicionamiento **static**

El valor `static` es el **valor predeterminado** para la propiedad `position` de todos los elementos HTML. Si no especificas explícitamente un valor diferente para `position`, un elemento será `static` por defecto.

Características Principales:

1. **Flujo Normal del Documento:** Los elementos con `position: static;` se renderizan en el orden en que aparecen en el flujo normal del documento HTML.
 - Los **elementos de bloque** (`<div>`, `<p>`, `<h1>`, etc.) ocupan todo el ancho disponible de su contenedor y se apilan verticalmente, uno debajo del otro.
 - Los **elementos en línea** (``, `<a>`, ``, ``, etc.) fluyen horizontalmente, uno tras otro, dentro de su bloque contenedor, y pasan a la siguiente línea si no hay suficiente espacio horizontal.
2. **Ignora Propiedades de Desplazamiento:** Las propiedades `top`, `right`, `bottom`, y `left` **no tienen ningún efecto** en un elemento con `position: static;`. Cualquier valor que les asignes será ignorado.

```
.mi-elemento-static {  
  position: static; /* O simplemente no declarar 'position' */  
  top: 50px; /* IGNORADO */  
  left: 100px; /* IGNORADO */  
  background-color: lightcoral;  
  padding: 10px;  
}
```

El elemento `.mi-elemento-static` se mostrará en su lugar normal en el flujo, sin ser desplazado 50px desde arriba o 100px desde la izquierda.

3. **Ignora `z-index`:** La propiedad `z-index`, que controla el orden de apilamiento de elementos superpuestos, **tampoco tiene efecto** en elementos con `position: static;`. Los elementos estáticos siempre se dibujan en el orden en que aparecen en el HTML, detrás de cualquier elemento posicionado (relative, absolute, fixed, sticky).
4. **No crea Contexto de Posicionamiento:** Un elemento con `position: static;` **no** actúa como un contenedor de referencia para los elementos hijos que tengan `position: absolute;`. Si un elemento absoluto está dentro de un elemento estático, buscará un ancestro posicionado más arriba en el árbol DOM, o se posicionará respecto al `<html>` si no encuentra ninguno.

¿Cuándo se usa `position: static;` ?

- **Es el comportamiento por defecto:** No necesitas declararlo a menos que necesites *restablecer* explícitamente el posicionamiento de un elemento que previamente tenía otro valor de `position` (por ejemplo, en un contexto de diseño responsive donde un elemento es `fixed` en pantallas grandes pero `static` en móviles).
- **Para elementos que deben seguir el flujo normal:** La mayoría de los elementos de una página (párrafos, encabezados, listas, imágenes estándar) usarán el posicionamiento estático porque simplemente deben aparecer en el orden natural del contenido.

Ejemplo Visual:

```
<div style="border: 1px solid blue; padding: 5px;">
  Contenedor Padre (static por defecto)
  <p style="background-color: lightyellow; margin: 5px 0;">Párrafo 1 (static)</p>
  <div style="background-color: lightblue; padding: 10px; position: static; top: -20px; left: 30px;">
    Div Estático (top/left ignorados)
  </div>
  <p style="background-color: lightgreen; margin: 5px 0;">Párrafo 2 (static)</p>
</div>
```

Resultado: El "Div Estático" aparecerá directamente debajo del "Párrafo 1" y encima del "Párrafo 2", siguiendo el flujo normal de bloques. Las propiedades `top: -20px;` y `left: 30px;` serán completamente ignoradas.

Conclusión:

`position: static;` es el modo de posicionamiento fundamental y predeterminado en CSS. Los elementos estáticos siguen el flujo normal del documento, no pueden ser desplazados con `top/right/bottom/left`, no responden a `z-index`, y no sirven como referencia para elementos con `position: absolute`. Entender `static` es clave para saber por qué y cuándo necesitamos usar los otros valores de `position` para lograr layouts más complejos.
