

33 CSS: Introducción al Posicionamiento CSS (`position`)

Hasta ahora, hemos visto cómo los elementos se colocan en la página siguiendo el **flujo normal del documento**: los elementos de bloque ocupan su propio espacio y se apilan verticalmente, mientras que los elementos en línea fluyen horizontalmente uno tras otro, como las palabras en un párrafo. También hemos visto cómo `float` puede sacar elementos de este flujo para permitir que el contenido fluya a su alrededor.

Sin embargo, a menudo necesitamos un control mucho más preciso sobre dónde se coloca un elemento, independientemente del flujo normal. Queremos poder superponer elementos, fijarlos en la ventana del navegador, o moverlos ligeramente respecto a su posición original. Para esto, utilizamos la propiedad `position` de CSS.

La Propiedad `position`

La propiedad `position` especifica el **método de posicionamiento** utilizado para un elemento. Establecer un valor distinto al predeterminado (`static`) permite usar las **propiedades de desplazamiento** (`top` , `right` , `bottom` , `left`) para mover el elemento desde su ubicación normal o en relación con un ancestro posicionado o la ventana del navegador.

Valores Principales de `position` :

La propiedad `position` puede tomar varios valores, cada uno con un comportamiento distinto. Los principales son:

1. `static` : Es el valor **predeterminado**. El elemento se posiciona según el **flujo normal del documento**. Las propiedades `top` , `right` , `bottom` , `left` y `z-index` **no tienen ningún efecto** en elementos con `position: static` .
2. `relative` : El elemento se posiciona según el flujo normal, pero luego **se desplaza** con respecto a su **posición original** (`static`) usando las propiedades `top` , `right` , `bottom` , `left` . Es importante destacar que el espacio que ocupaba originalmente el elemento en el flujo normal **se reserva**, y otros elementos no se moverán para llenar ese espacio vacío. Además, `position: relative` crea un **contexto de posicionamiento** para sus elementos descendientes que tengan `position: absolute` .
3. `absolute` : El elemento **se elimina completamente del flujo normal** del documento (no deja espacio reservado). Se posiciona con respecto a su **ancestro posicionado más cercano** (un ancestro cuya propiedad `position` sea distinta de `static`). Si no existe un ancestro posicionado, se posiciona con respecto al bloque contenedor inicial, que generalmente es el elemento `<html>` (efectivamente, la ventana del navegador). Las propiedades `top` , `right` , `bottom` , `left` determinan la distancia desde los bordes de este ancestro posicionado.
4. `fixed` : El elemento **se elimina completamente del flujo normal**. Se posiciona con respecto a la **ventana gráfica del navegador (viewport)**. Esto significa que permanecerá en el mismo lugar en la pantalla **incluso si se hace scroll** en la página. Las propiedades `top` , `right` , `bottom` , `left` determinan la distancia desde los bordes de la ventana gráfica. También crea un nuevo contexto de apilamiento (stacking context).
5. `sticky` : Es un híbrido entre `relative` y `fixed` . El elemento se trata como `position: relative` hasta que su bloque contenedor cruza un umbral especificado (definido por `top` , `right` , `bottom` , o `left`) durante

el scroll. A partir de ese punto, se "pega" y se comporta como `position: fixed` dentro de su contenedor directo o un ancestro con scroll.

Las Propiedades de Desplazamiento (`top` , `right` , `bottom` , `left`)

Estas propiedades solo funcionan en elementos cuya `position` **no** sea `static` . Especifican la distancia a la que se moverá el elemento o a qué distancia se colocará de los bordes de su referencia (posición original, ancestro posicionado o viewport).

- `top` : Distancia desde el borde superior de la referencia.
- `bottom` : Distancia desde el borde inferior de la referencia.
- `left` : Distancia desde el borde izquierdo de la referencia.
- `right` : Distancia desde el borde derecho de la referencia.

Se pueden usar valores positivos (alejan del borde) o negativos (acercan o incluso superan el borde). No es necesario definir las cuatro; a menudo se definen solo dos (p.ej., `top` y `left` , o `bottom` y `right`).

El Contexto de Posicionamiento (Containing Block)

Este es un concepto crucial, especialmente para `position: absolute` . El "bloque contenedor" o "contexto de posicionamiento" es el rectángulo de referencia contra el cual se posiciona un elemento absoluto. Como se mencionó:

- Para `position: absolute` , es el **ancestro más cercano con `position` diferente de `static` .**
- Si no hay tal ancestro, es el bloque contenedor inicial (generalmente el `<html>`).
- Para `position: fixed` , es la **ventana gráfica (viewport)**.
- Para `position: relative` , el desplazamiento es relativo a **su propia posición normal**.
- Para `position: sticky` , depende de si está en modo relativo o fijo.

El Contexto de Apilamiento y `z-index`

Cuando los elementos se posicionan fuera del flujo normal, pueden superponerse. La propiedad `z-index` (que veremos en detalle más adelante) controla el orden de apilamiento de estos elementos posicionados (aquellos con `position` diferente de `static`). Un `z-index` mayor coloca un elemento por encima de otro con un `z-index` menor.

Próximos Pasos:

En las siguientes secciones, exploraremos en detalle cada uno de los valores de `position` (`static` , `relative` , `absolute` , `fixed` , `sticky`), veremos ejemplos prácticos de su uso y comprenderemos mejor cómo interactúan entre sí y con las propiedades de desplazamiento.
