

38 CSS: Posicionamiento **sticky**

El posicionamiento pegajoso (`position: sticky;`) es un valor relativamente moderno que ofrece una combinación inteligente entre el comportamiento del posicionamiento relativo (`relative`) y el fijo (`fixed`). Un elemento `sticky` se trata inicialmente como `relative` dentro del flujo normal, pero se "pega" en una posición fija respecto a su contenedor de scroll más cercano una vez que la página se desplaza más allá de un punto umbral definido.

Características Principales:

1. Comportamiento Híbrido:

- **Modo Relativo (Inicial):** Al principio, el elemento ocupa su espacio normal en el flujo del documento, tal como lo haría un elemento `relative` . Se desplaza junto con el resto del contenido al hacer scroll.
- **Modo Fijo/Pegajoso (Activado):** Cuando el scroll de la página hace que el elemento alcance el umbral definido por una de las propiedades `top` , `right` , `bottom` o `left` (relativo a su contenedor de scroll), el elemento deja de moverse con el scroll y se fija en esa posición dentro de los límites de su contenedor.

2. Necesidad de `top` , `right` , `bottom` , o `left` :

Para que `position: sticky;` funcione, es **obligatorio** especificar al menos una de estas propiedades de desplazamiento. Esta propiedad no mueve el elemento inicialmente (como haría en `relative`), sino que define el **punto de activación** (el umbral) en el que el elemento debe empezar a pegarse.

- `position: sticky; top: 0;` : El elemento se pegará al borde superior de su contenedor de scroll cuando el scroll haga que el borde superior del elemento intente pasar por encima del borde superior del contenedor de scroll.
- `position: sticky; bottom: 10px;` : Se pegará a 10px del borde inferior del contenedor de scroll cuando el scroll haga que su borde inferior intente moverse más allá de esa posición.

3. Contenido Dentro de un Ancestro con Scroll:

Esta es la diferencia fundamental con `position: fixed;` . Un elemento `sticky` está **limitado a los límites de su ancestro más cercano que genera un scroll** (un elemento con `overflow: hidden` , `scroll` , `auto` , o el propio viewport si no hay otro). El elemento `sticky` se pegará en su posición definida (`top` , `bottom` , etc.) *mientras esté dentro de la vista de este ancestro*. Una vez que el final del ancestro con scroll desaparece de la pantalla (al seguir haciendo scroll), el elemento `sticky` se "despegará" y se moverá junto con el final de su contenedor. No puede salirse de su contenedor padre con scroll.

4. Permanece en el Flujo (Conceptualmente):

Aunque cambia su comportamiento, no se elimina del flujo de la misma manera que `absolute` o `fixed` . Su espacio original sigue influyendo en el layout de los elementos que lo rodean, especialmente antes de alcanzar el estado "pegajoso".

5. Funciona con `z-index` :

Se puede usar `z-index` para controlar su nivel de apilamiento, especialmente útil cuando está en estado pegajoso y puede superponerse a otro contenido.

6. Crea Contexto de Apilamiento:

Cuando el elemento está efectivamente "pegado", crea un contexto de apilamiento.

Ejemplo Visual (Cabecera de Sección Sticky):

- **HTML:**

```

<div class="container">
  <h1>Página con Contenido y Sticky Header</h1>
  <p>Algo de contenido inicial...</p>

  <div class="section" id="section1">
    <h2 class="sticky-header">Cabecera Sticky Sección 1</h2>
    <div class="content">
      <p>Contenido de la sección 1...</p>
      <p>Más contenido para rellenar y generar scroll...</p>
      <p>...</p><p>...</p><p>...</p><p>...</p>
      <p>Final Sección 1.</p>
    </div>
  </div>

  <div class="section" id="section2">
    <h2 class="sticky-header">Cabecera Sticky Sección 2</h2>
    <div class="content">
      <p>Contenido de la sección 2...</p>
      <p>Más contenido para rellenar...</p>
      <p>...</p><p>...</p><p>...</p><p>...</p>
      <p>Final Sección 2.</p>
    </div>
  </div>

  <p>Algo de contenido final...</p>
</div>

```

- **CSS:**

```

body {
  font-family: sans-serif;
  margin: 0;
}

.container {
  max-width: 800px;
  margin: 0 auto;
  padding: 20px;
}

.section {
  margin-bottom: 30px;
  border: 1px solid #ccc;
  background-color: #f9f9f9;
}

.sticky-header {

```

```

position: sticky; /* Activamos el posicionamiento pegajoso */
top: 0; /* Se pegará al borde superior del viewport (o contenedor con scroll) */
background-color: #333;
color: white;
padding: 10px 20px;
margin: 0; /* Quitamos márgenes por defecto del h2 */
z-index: 10; /* Para que esté sobre el contenido de la sección */
}

.content {
padding: 0 20px 20px 20px; /* Padding solo para el contenido */
}

.content p { line-height: 1.6; }

```

- **Resultado:**

1. Mientras haces scroll, las cabeceras (`.sticky-header`) se moverán normalmente con su sección.
2. Cuando el borde superior de una cabecera sticky esté a punto de desaparecer por la parte superior de la ventana (el viewport, en este caso, es el contenedor de scroll), esa cabecera se **pegará** en `top: 0`.
3. Permanecerá pegada en la parte superior mientras haces scroll *dentro* del contenido de su propia sección (`.section`).
4. Tan pronto como el *final* de su sección padre (`.section`) alcance la cabecera pegada (es decir, cuando toda la sección está a punto de salir de la vista por arriba), la cabecera se "despegará" y se moverá hacia arriba junto con el resto de su sección, dejando espacio para que la siguiente cabecera sticky (si la hay) tome su lugar cuando llegue al umbral `top: 0`.

Casos de Uso Comunes para `position: sticky;` :

- **Cabeceras de Sección:** Mantener visible el título de la sección actual mientras se recorre su contenido.
- **Cabeceras de Tabla (`<thead>`):** Mantener los encabezados de las columnas visibles mientras se desplaza por las filas de una tabla larga.
- **Barras Laterales Contextuales:** Una barra lateral que permanece visible solo mientras se navega por una sección específica de la página.
- **Elementos de Formulario Importantes:** Mantener visible un botón de "Guardar" o resumen mientras se completa un formulario largo.

Consideraciones:

- **Soporte de Navegadores:** Aunque el soporte es muy bueno en navegadores modernos, siempre es prudente verificar con `caniuse.com` si necesitas compatibilidad con versiones muy antiguas.
- **Dependencia del Ancestro con Scroll:** Su comportamiento depende totalmente de tener un ancestro que permita el scroll. Si el ancestro no tiene `overflow` configurado o no es lo suficientemente grande como para generar scroll, `sticky` se comportará simplemente como `relative`.

- **Necesidad de `top/bottom/left/right`** : Sin definir al menos una de estas propiedades, `sticky` no tendrá efecto.

Conclusión:

`position: sticky;` ofrece una solución elegante y eficiente para crear elementos que se mantienen visibles contextualmente durante el scroll, sin la complejidad de JavaScript y sin salirse de los límites de su contenedor padre. Es ideal para mejorar la usabilidad en páginas con secciones largas o tablas de datos extensas.
