

40 CSS: Cabecera `fixed` VS Cabecera `sticky`

Tanto `position: fixed` como `position: sticky` se pueden utilizar para crear cabeceras (o cualquier otro elemento) que permanezcan visibles mientras el usuario se desplaza por la página. Sin embargo, tienen diferencias fundamentales en su comportamiento, su referencia de posicionamiento y sus casos de uso ideales.

1. Cabecera con `position: fixed`

- **Comportamiento:** La cabecera se **elimina completamente del flujo normal** del documento desde el principio. Se posiciona **respecto a la ventana gráfica (viewport)** y permanece **fija en la misma posición** (generalmente `top: 0; left: 0; width: 100%;`) sin importar cuánto se desplace la página.
- **Referencia:** Viewport.
- **Impacto en el Flujo:** Al ser eliminada del flujo, el contenido que sigue en el HTML comenzará *detrás* de la cabecera fija si no se toman medidas.
- **Manejo de Superposición:** Es **necesario** añadir un `padding-top` (o `margin-top`) al `body` o al contenedor principal del contenido que sea igual o mayor que la altura de la cabecera fija. Esto "empuja" el inicio del contenido hacia abajo para que no quede oculto detrás de la cabecera.
- **Limitación:** Ninguna respecto a contenedores. Siempre estará fija en la ventana.
- **CSS Ejemplo (Básico):**

```
body {  
  padding-top: 60px; /* Espacio para la cabecera fija */  
}  
  
.header-fixed {  
  position: fixed;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 60px; /* Altura de la cabecera */  
  background-color: #333;  
  color: white;  
  z-index: 100;  
  /* ... otros estilos ... */  
}
```

- **Pros:**
 - Comportamiento simple y predecible: siempre está ahí.
 - Ideal para cabeceras globales que deben ser accesibles desde cualquier punto de la página.
- **Contras:**

- Requiere ajustar el padding/margen del contenido principal para evitar superposiciones.
- Puede sentirse un poco desconectada del contenido ya que nunca se mueve con él.
- Ocupa espacio vertical permanentemente, lo cual puede ser problemático en pantallas pequeñas.

2. Cabecera con `position: sticky`

- **Comportamiento:** La cabecera comienza como parte del **flujo normal** (similar a `relative`). Solo se vuelve "pegajosa" (actúa como `fixed`) cuando el scroll alcanza el umbral definido por `top`, `right`, `bottom`, o `left`.
- **Referencia:** Inicialmente el flujo; cuando está pegada, se posiciona respecto a su **ancestro con scroll más cercano**.
- **Impacto en el Flujo:** Inicialmente ocupa su espacio normal. El contenido siguiente fluye naturalmente después de ella. No requiere un ajuste inicial de `padding` en el body *solo* por la cabecera.
- **Manejo de Superposición:** Una vez que se vuelve pegajosa, puede superponer el contenido siguiente si no se gestiona con `z-index` adecuado. Sin embargo, no oculta el contenido *inicial* porque empieza en el flujo.
- **Limitación:** Está **limitada por los bordes de su ancestro con scroll**. Si tienes una cabecera sticky dentro de una sección `<div class="scrollable-section">`, la cabecera solo será sticky *mientras esa sección esté visible* y no se saldrá de ella. Si el ancestro es el `body` / `html`, se comportará de forma similar a `fixed` pero solo después de alcanzar el umbral.
- **CSS Ejemplo (Básico):**

```
/* No se necesita padding-top inicial en body solo para esto */

.header-sticky {
  position: sticky;
  top: 0; /* Se pegará arriba cuando se alcance este punto */
  background-color: #555;
  color: white;
  height: 60px;
  z-index: 100;
  /* ... otros estilos ... */
}
```

- **Pros:**
 - Integración más fluida: la cabecera empieza con su contenido.
 - No requiere ajustes iniciales de `padding` en el body.
 - Ideal para cabeceras de secciones específicas que solo deben ser persistentes mientras se visualiza esa sección.
 - Permite tener múltiples elementos `sticky` en la página (ej. cabeceras de diferentes secciones que se reemplazan al hacer scroll).
- **Contras:**

- El comportamiento depende de la estructura del DOM y de los contenedores con scroll.
- Puede ser ligeramente menos predecible si no se entiende bien el concepto de "ancestro con scroll".
- Si se desea que *siempre* esté visible desde el inicio (como una cabecera global), `fixed` puede ser más directo.

Resumen de Diferencias Clave:

Característica	<code>position: fixed</code>	<code>position: sticky</code>
Referencia	Viewport (Ventana del Navegador)	Ancestro con Scroll más cercano
Inicio	Fuera del flujo, fijo desde el inicio	En flujo (como <code>relative</code>), se pega después
Siempre Visible	Sí	Solo después de alcanzar <code>top</code> / <code>bottom</code> etc.
Sale del Contenedor	Sí (ignora contenedores)	No (limitado por su ancestro con scroll)
Padding Inicial	Generalmente Necesario (en <code>body</code>)	Generalmente No Necesario

¿Cuándo ElegirCuál?

- Usa `position: fixed` si:
 - Necesitas que la cabecera esté **absolutamente siempre visible**, desde el momento en que carga la página, sin importar dónde esté el usuario en el scroll.
 - Es una cabecera **global** para todo el sitio.
 - No te importa (o prefieres) que esté completamente desacoplada del flujo del contenido.
- Usa `position: sticky` si:
 - Quieres que la cabecera **aparezca inicialmente en su lugar** y solo se fije después de un poco de scroll.
 - La cabecera pertenece a una **sección específica** y solo debe permanecer visible mientras esa sección está en pantalla.
 - Quieres **múltiples cabeceras** en la página que se "peguen" y reemplacen a medida que el usuario navega por diferentes secciones.
 - Prefieres evitar añadir `padding` al `body` solo para acomodar la cabecera.

Conclusión:

Ambas opciones son válidas para crear cabeceras persistentes. `fixed` ofrece una persistencia total y simple respecto al viewport, a costa de requerir un ajuste manual del layout inicial. `sticky` proporciona una experiencia más integrada y contextual, ideal para elementos que deben fijarse solo en ciertas partes del scroll y dentro de ciertos límites, pero su comportamiento depende más de la estructura HTML circundante. La elección dependerá de los requisitos específicos de tu diseño y la experiencia de usuario deseada.