

41 CSS: Efecto de Slides (Diapositivas) con position: sticky

Una aplicación creativa y visualmente atractiva de position: sticky es simular un efecto de "diapositivas" o "slides" a pantalla completa, donde cada sección de la página parece deslizarse sobre la anterior a medida que el usuario hace scroll.

El Concepto:

La idea es tener varias secciones (slides) que ocupen, como mínimo, el alto completo de la ventana gráfica (100vh). A cada una de estas secciones se le aplica position: sticky y top: 0. Cuando el usuario hace scroll, la sección actual sube hasta que la siguiente sección llega al borde superior del viewport. En ese momento, la nueva sección se vuelve sticky (se pega en top: 0), cubriendo efectivamente a la sección anterior, dando la ilusión de que se deslizó sobre ella.

Estructura HTML Necesaria:

Necesitamos una estructura simple con múltiples secciones que representarán las "slides". A menudo, estas secciones estarán directamente dentro del <body> o de un contenedor principal.

```
<body>
<section class="slide slide-1">
  <h1>Slide 1</h1>
  <p>Contenido de la primera diapositiva...</p>
</section>

<section class="slide slide-2">
  <h1>Slide 2</h1>
  <p>Contenido de la segunda diapositiva...</p>
</section>

<section class="slide slide-3">
  <h1>Slide 3</h1>
  <p>Contenido de la tercera diapositiva...</p>
</section>

<!-- Puedes añadir más slides si es necesario -->

</body>
```

CSS Clave para el Efecto:

Las reglas CSS fundamentales para lograr este efecto son:

```
body {
  margin: 0; /* Eliminar margen por defecto del body */
  font-family: sans-serif;
}
```

```

.slide {
  position: sticky; /* La clave del efecto */
  top: 0; /* Pegar al borde superior cuando se alcance */
  min-height: 100vh; /* ¡Fundamental! Cada slide ocupa al menos toda la altura del viewport */
  box-sizing: border-box; /* Para que el padding no aumente la altura total más allá de 100vh si l
o usamos */
  padding: 40px; /* Espacio interno para el contenido */

  /* Estilos visuales para diferenciarlas (ejemplos) */
  display: flex; /* Opcional: para centrar contenido fácilmente */
  flex-direction: column; /* Opcional */
  justify-content: center; /* Opcional */
  align-items: center; /* Opcional */
  text-align: center; /* Opcional */
  color: white; /* Ejemplo */
}

/* Colores de fondo distintos para cada slide */
.slide-1 {
  background-color: #4a90e2; /* Azul */
  /* z-index: 1; (Opcional, el orden HTML suele bastar) */
}

.slide-2 {
  background-color: #50e3c2; /* Verde agua */
  /* z-index: 2; */
}

.slide-3 {
  background-color: #b8e986; /* Verde claro */
  /* z-index: 3; */
}

/* Puedes añadir estilos específicos para el contenido de cada slide */
.slide h1 {
  font-size: 3em;
  margin-bottom: 20px;
}

```

¿Cómo Funciona Exactamente?

1. **Altura 100vh:** Cada `.slide` ocupa toda la altura de la ventana. Esto asegura que cuando una slide se vuelve sticky en `top: 0`, cubre completamente lo que había debajo. También significa que el usuario debe hacer scroll una altura considerable (la altura de un viewport) para llegar al inicio de la siguiente slide.
2. **position: sticky; top: 0;:**
 - Inicialmente, `slide-1` está visible.

- Al hacer scroll, slide-1 se mueve hacia arriba.
- Cuando el borde superior de slide-2 llega al borde superior del viewport (top: 0), slide-2 se vuelve sticky. Ahora slide-2 está pegada arriba, cubriendo a slide-1.
- El usuario sigue haciendo scroll "a través" de la altura de slide-2 (que es 100vh). slide-2 permanece pegada arriba.
- Cuando el borde superior de slide-3 llega a top: 0, slide-3 se vuelve sticky, cubriendo a slide-2.
- Este proceso se repite para cada slide. El orden natural del HTML asegura que la slide posterior se dibuje encima de la anterior cuando ambas están en top: 0 (o puedes usar z-index si necesitas un control más explícito, aunque no suele ser necesario para este efecto básico).

Ventajas de este Enfoque:

- **CSS Puro:** No requiere JavaScript para el efecto de deslizamiento/apilamiento principal.
- **Rendimiento:** Generalmente es más performante que las soluciones basadas en JavaScript que manipulan constantemente la posición durante el scroll.
- **Relativamente Simple:** La lógica CSS central es concisa.

Consideraciones:

- **Contenido Largo:** Si el contenido dentro de una slide es *más* alto que 100vh, el usuario tendrá que hacer scroll dentro de esa slide mientras está pegada antes de poder pasar a la siguiente. `min-height: 100vh;` permite este comportamiento. Si usas `height: 100vh;` y el contenido se desbordara, necesitarías manejar el overflow dentro de la slide.
- **Accesibilidad:** Asegúrate de que el contenido siga siendo accesible y que la navegación (especialmente con teclado) funcione correctamente.
- **Contexto de Scroll:** Este efecto funciona mejor cuando el `<body>` o `<html>` es el contenedor de scroll principal. Si las slides están dentro de un div con `overflow: auto` y altura limitada, el efecto ocurrirá dentro de ese div.