

37 CSS: Posicionamiento **fixed**

El posicionamiento fijo (`position: fixed;`) es muy similar al posicionamiento absoluto en varios aspectos, pero con una diferencia clave en su referencia de posicionamiento: se posiciona con respecto a la **ventana gráfica (viewport)** del navegador, no a un ancestro posicionado.

Características Principales:

1. **Eliminado del Flujo Normal:** Al igual que `position: absolute;`, los elementos con `position: fixed;` **se eliminan completamente del flujo normal del documento**. No reservan espacio, y otros elementos se comportarán como si el elemento fijo no existiera en el flujo.
2. **Posicionamiento Respecto al Viewport:** Esta es la característica distintiva. Un elemento fijo se posiciona utilizando las propiedades `top`, `right`, `bottom`, `left` con respecto a los **bordes de la ventana gráfica del navegador**.
3. **Permanece Fijo Durante el Scroll:** Debido a que se posiciona respecto al viewport, un elemento con `position: fixed;` **permanecerá en el mismo lugar en la pantalla** incluso cuando el usuario desplace (haga scroll) la página hacia arriba o hacia abajo. No se moverá con el resto del contenido.
4. **Efecto de `top`, `right`, `bottom`, `left`:** Determinan la distancia entre el borde correspondiente del elemento fijo y el borde correspondiente de la ventana gráfica.
 - `top: 0; left: 0; width: 100%;` : Crearía una barra fija en la parte superior de la ventana, ocupando todo su ancho.
 - `bottom: 20px; right: 20px;` : Colocaría un elemento fijo en la esquina inferior derecha de la ventana, a 20px de cada borde.
5. **Comportamiento de Tamaño (Shrink-to-fit):** Similar a `absolute`, tiende a encogerse para ajustarse a su contenido si no se especifican `width` / `height` o propiedades de desplazamiento opuestas.
6. **Funciona con `z-index`:** Es crucial para los elementos fijos, ya que a menudo se superponen a todo lo demás. `z-index` permite controlar si el elemento fijo aparece por encima o por debajo de otros elementos (incluidos otros elementos fijos o absolutos).
7. **Crea Contexto de Apilamiento (Stacking Context):** Un elemento con `position: fixed;` crea su propio contexto de apilamiento.
8. **Puede Crear Contexto de Posicionamiento:** Aunque menos común que con `relative` o `absolute`, un elemento `fixed` también puede servir como bloque contenedor de referencia para descendientes con `position: absolute;`.

Ejemplo Visual:

- **HTML:**

```
<body>
<header class="cabecera-fija">
  Cabecera Fija
</header>
```

```

<main class="contenido-principal">
  <h1>Contenido Principal</h1>
  <p>Este es un párrafo largo para generar scroll...</p>
  <!-- Repetir el párrafo muchas veces →
  <p>...</p><p>...</p><p>...</p><p>...</p><p>...</p><p>...</p><p>...</p>
  <p>...</p><p>...</p><p>...</p><p>...</p><p>...</p><p>...</p><p>...</p>
  <p>Fin del contenido.</p>
</main>

<div class="boton-fijo">
  Arriba
</div>
</body>

```

- **CSS:**

```

body {
  /* Añadimos padding-top para evitar que el contenido
  inicial quede oculto DEBAJO de la cabecera fija */
  padding-top: 60px; /* Debe ser al menos la altura de la cabecera */
  font-family: sans-serif;
}

.cabecera-fija {
  position: fixed; /* Posicionamiento fijo */
  top: 0; /* Pegado a la parte superior del viewport */
  left: 0; /* Pegado a la parte izquierda del viewport */
  width: 100%; /* Ocupa todo el ancho */
  height: 50px; /* Altura de la cabecera */
  background-color: #333;
  color: white;
  padding: 5px 20px; /* Padding interno */
  box-sizing: border-box; /* Incluir padding en height/width */
  z-index: 1000; /* Asegura que esté por encima de la mayoría de elementos */
  line-height: 40px; /* Para centrar verticalmente el texto (si es una línea) */
  text-align: center;
}

.contenido-principal {
  width: 80%;
  margin: 20px auto; /* Centrar el contenido */
  padding: 20px;
  border: 1px solid #ccc;
  background-color: #f9f9f9;
}

.contenido-principal p { line-height: 1.6; margin-bottom: 15px; }

```

```
.boton-fijo {
  position: fixed; /* Posicionamiento fijo */
  bottom: 20px; /* 20px desde el borde inferior del viewport */
  right: 20px; /* 20px desde el borde derecho del viewport */
  background-color: dodgerblue;
  color: white;
  padding: 10px 15px;
  border-radius: 5px;
  cursor: pointer;
  z-index: 1001; /* Incluso por encima de la cabecera */
}
```

- **Resultado:**

1. La `cabecera-fija` negra permanecerá pegada en la parte superior de la ventana del navegador sin importar cuánto se desplace la página.
2. El `contenido-principal` comenzará visiblemente *debajo* de la cabecera gracias al `padding-top` aplicado al `body`. Sin ese padding, las primeras líneas del contenido quedarían ocultas detrás de la cabecera fija.
3. El `boton-fijo` azul permanecerá en la esquina inferior derecha de la ventana mientras se hace scroll.

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

- **Cabeceras y Pies de Página Fijos:** Barras de navegación o información que siempre deben estar visibles.
- **Barras Laterales Fijas:** Menús o paneles que permanecen en su lugar mientras el contenido principal se desplaza.
- **Botones de "Volver Arriba" (`Back to Top`).**
- **Banners de Cookies o Notificaciones:** Mensajes que se muestran fijos hasta que el usuario interactúa con ellos.
- **Fondos de Ventanas Modales:** Un overlay semitransparente que cubre toda la ventana detrás de un diálogo modal.

Consideraciones Importantes:

- **Obscurecimiento de Contenido:** Los elementos fijos pueden tapar contenido, especialmente en pantallas pequeñas (móviles). Es crucial asegurarse de que haya suficiente espacio o de que el diseño se adapte (por ejemplo, añadiendo `padding` al `body` o al contenedor principal como en el ejemplo).
- **Accesibilidad:** Asegúrate de que los elementos fijos no impidan el acceso a otras partes de la página, especialmente con navegación por teclado o lectores de pantalla.
- **Contexto de Apilamiento:** El `z-index` es vital para controlar qué aparece encima de qué.

Conclusión:

`position: fixed;` es ideal para elementos que deben permanecer visibles y en la misma posición en la pantalla independientemente del scroll de la página. Saca el elemento del flujo normal y lo posiciona respecto a la ventana gráfica. Es fundamental para crear interfaces de usuario

persistentes como cabeceras, pies de página o botones de acción fijos, pero requiere cuidado para no obstruir el contenido principal.
