

39 CSS: Propiedad **z-index**

Cuando utilizamos valores de `position` distintos de `static` (`relative`, `absolute`, `fixed`, `sticky`), los elementos pueden salir del flujo normal y, como resultado, superponerse unos a otros. La propiedad `z-index` nos permite controlar **el orden de apilamiento** de estos elementos posicionados a lo largo del eje Z (el eje imaginario que va desde la pantalla hacia el espectador).

¿Qué es el Eje Z y el Orden de Apilamiento?

Imagina que tu página web no es plana, sino que tiene profundidad. El eje Z representa esta profundidad. Un elemento con un `z-index` más alto se considera que está "más cerca" del espectador (más arriba en la pila), y por lo tanto, se dibujará encima de los elementos con un `z-index` más bajo.

¿Cómo Funciona `z-index` ?

1. **Valores:** `z-index` acepta valores enteros (positivos, negativos o cero) o la palabra clave `auto`.
 - **Enteros:** `z-index: 1;`, `z-index: 99;`, `z-index: -1;`, `z-index: 0;`
 - **auto:** Es el valor por defecto. El elemento se apila según su orden en el HTML dentro de su contexto de apilamiento. No establece un nuevo contexto de apilamiento por sí mismo si la `position` es `relative` o `static` y no tiene otras propiedades que lo creen.
2. **Regla de Apilamiento:**
 - Un elemento con un valor de `z-index` mayor se mostrará **encima** de un elemento con un valor de `z-index` menor, **siempre que pertenezcan al mismo contexto de apilamiento**.
 - Si dos elementos posicionados tienen el **mismo** `z-index`, su orden de apilamiento se determina por su **orden de aparición en el código HTML**. El elemento que aparece más tarde en el HTML se dibujará encima.

Prerrequisito Fundamental: ¡Solo para Elementos Posicionados!

Es crucial recordar que la propiedad `z-index` **solo tiene efecto en elementos cuya propiedad `position` tenga un valor distinto de `static`**. Si intentas aplicar `z-index` a un elemento con `position: static;` (el valor por defecto), será ignorado.

```
.elemento-no-posicionado {
  /* position: static; (implícito) */
  z-index: 100; /* ESTO NO TENDRÁ EFECTO */
  background-color: red;
}

.elemento-posicionado {
  position: relative; /* O absolute, fixed, sticky */
  z-index: 10; /* ESTO SÍ TENDRÁ EFECTO */
  background-color: blue;
}
```

El Contexto de Apilamiento (Stacking Context)

Este es un concepto fundamental para entender `z-index` correctamente. No todos los elementos de una página compiten en el mismo "nivel" de apilamiento global. Ciertos elementos crean un **contexto de apilamiento local**.

- **¿Qué es?** Un contexto de apilamiento es un grupo de elementos que se mueven juntos en la pila, como una unidad. Dentro de un contexto de apilamiento, los valores de `z-index` solo se comparan entre los elementos *dentro* de ese contexto. El contexto de apilamiento completo se ordena luego como una sola entidad en el contexto de apilamiento padre.
- **¿Qué crea un Contexto de Apilamiento?**
 - El elemento raíz (`<html>`).
 - Elementos con `position: absolute;` o `position: relative;` y un valor de `z-index` distinto de `auto` .
 - Elementos con `position: fixed;` o `position: sticky;` .
 - Elementos que son hijos de un contenedor `flex` o `grid` y tienen un `z-index` distinto de `auto` .
 - Elementos con `opacity` menor que 1.
 - Elementos con `transform` , `filter` , `perspective` , `clip-path` , `mask` aplicados (con valores distintos a los iniciales).
 - Otros casos menos comunes (`will-change` , `contain` , etc.).
- **Implicación Clave:** Un elemento con `z-index: 9999;` dentro de un contexto de apilamiento con `z-index: 1;` **no** aparecerá por encima de un elemento con `z-index: 2;` que esté *fuera* de ese contexto y pertenezca a un contexto superior. El `z-index: 9999` solo "gana" dentro de su propio grupo (`z-index: 1`).

Ejemplo Visual:

- **HTML:**

```
<div class="contenedor">
  <div class="caja caja1">Caja 1 (z-index: 1)</div>
  <div class="caja caja2">Caja 2 (z-index: 2)</div>
  <div class="caja caja3">Caja 3 (z-index: auto/0 - default)</div>
  <div class="caja caja4">Caja 4 (z-index: -1)</div>
</div>
```

- **CSS:**

```
.contenedor {
  position: relative; /* Padre crea contexto */
  height: 250px;
  border: 1px dashed grey;
  margin-top: 30px;
}

.caja {
  position: absolute; /* Todas son absolutas para superponerse */
  width: 150px;
  height: 150px;
  font-size: 18px;
```

```

color: white;
padding: 10px;
box-sizing: border-box;
text-align: center;
line-height: 130px; /* Aprox. centrado vertical */
}
.caja1 {
background-color: dodgerblue;
top: 20px; left: 20px;
z-index: 1;
}
.caja2 {
background-color: crimson;
top: 40px; left: 70px;
z-index: 2; /* Valor más alto, estará encima de caja1 y caja3 */
}
.caja3 {
background-color: forestgreen;
top: 60px; left: 120px;
/* z-index: auto; o z-index: 0; */ /* Se apila según HTML después de negativos, antes de p
ositivos > 0 */
}
.caja4 {
background-color: darkorange;
top: 80px; left: 170px;
z-index: -1; /* Valor negativo, estará debajo de todas las demás */
}

```

• Resultado:

1. La Caja 4 (naranja, `z-index: -1`) estará en el fondo.
2. Encima de ella estará la Caja 3 (verde, `z-index: auto/0`).
3. Encima de la Caja 3 estará la Caja 1 (azul, `z-index: 1`).
4. La Caja 2 (roja, `z-index: 2`) estará en la parte superior, superponiéndose a todas las demás.

Casos de Uso Comunes:

- Asegurar que los menús desplegables aparezcan sobre el resto del contenido.
- Mostrar ventanas modales o pop-ups por encima de un overlay y del contenido principal.
- Controlar la superposición de elementos en diseños complejos o animaciones.
- Asegurar que cabeceras o pies de página fijos/sticky estén por encima del contenido que se desplaza.

Buenas Prácticas:

- **Usa `z-index` solo cuando sea necesario:** No lo apliques a todos los elementos.
- **Gestiona los valores:** No uses números enormes al azar (como `99999`). Intenta mantener los valores lo más bajos posible y organízalos lógicamente por componente o sección. Usa

variables CSS si es posible para gestionarlos.

- **Comprende los Contextos de Apilamiento:** Si `z-index` no funciona como esperas, lo más probable es que estés lidiando con diferentes contextos de apilamiento. Identifica qué elementos los crean.
- **No olvides `position`** : Recuerda que `z-index` no funciona en elementos `static` .

Conclusión:

La propiedad `z-index` es la herramienta clave para controlar el orden de apilamiento vertical (eje Z) de los elementos **posicionados**. Funciona asignando números enteros que indican la "altura" en la pila, pero su comportamiento está intrínsecamente ligado al concepto de **contextos de apilamiento**.
