

# 27 CSS: Propiedad **overflow**

La propiedad **overflow** en CSS es fundamental para controlar qué sucede cuando el contenido de un elemento en bloque (como un **div**) es demasiado grande para caber dentro del área designada para ese elemento (su "caja"). Esto puede ocurrir si el contenido (texto, imágenes, otros elementos hijos) excede el **width** o **height** explícito del contenedor, o si el contenido simplemente no cabe de forma natural.

## Contexto de Aplicación:

**overflow** generalmente se aplica a **elementos de nivel de bloque** que tienen una **altura (height) o altura máxima (max-height) definida**. Si un elemento no tiene una altura restringida, normalmente crecerá para acomodar su contenido, y **overflow** no tendrá mucho efecto en la dirección vertical (aunque sí puede afectar horizontalmente si el **width** está restringido y el contenido es muy ancho).

## Valores Principales de **overflow**:

### 1. **visible**:

- Este es el valor **predeterminado**.
- El contenido que desborda la caja del elemento **se muestra**, extendiéndose fuera de los límites del elemento. Puede superponerse a otros elementos cercanos en la página.
- **Ejemplo CSS:**

```
.caja-visible {  
  width: 200px;  
  height: 100px;  
  border: 1px solid red;  
  overflow: visible; /* Default, no es necesario declararlo */  
}
```

- **Resultado:** Si el texto o contenido dentro de **.caja-visible** ocupa más de 200×100 píxeles, se saldrá del borde rojo.

### 2. **hidden**:

- El contenido que desborda **se oculta (se recorta)** en los límites de la caja del elemento. No se proporcionan barras de desplazamiento para ver el contenido oculto.
- **Ejemplo CSS:**

```
.caja-oculta {  
  width: 200px;  
  height: 100px;  
  border: 1px solid blue;  
  overflow: hidden;  
}
```

- **Resultado:** Cualquier contenido que exceda los 200×100 píxeles será invisible. Solo se verá la parte del contenido que cabe dentro de la caja.

### 3. `scroll` :

- El contenido que desborda se oculta, pero el navegador **muestra barras de desplazamiento** (tanto horizontal como vertical) para permitir al usuario ver el resto del contenido.
- **Importante:** Las barras de desplazamiento se muestran **siempre**, incluso si el contenido cabe perfectamente dentro de la caja y no hay desbordamiento. Esto puede reservar espacio visual para las barras.
- **Ejemplo CSS:**

```
.caja-scroll {  
  width: 200px;  
  height: 100px;  
  border: 1px solid green;  
  overflow: scroll;  
}
```

- **Resultado:** La caja tendrá barras de desplazamiento visibles en la parte inferior y/o derecha, independientemente de si el contenido desborda o no. Si desborda, las barras serán funcionales.

### 4. `auto` :

- Este es a menudo el valor más útil. Es similar a `scroll`, pero el navegador **solo muestra las barras de desplazamiento si el contenido realmente desborda** la caja. Si el contenido cabe, no se mostrarán barras de desplazamiento.
- El navegador decide qué barras mostrar (horizontal, vertical o ambas) según sea necesario.
- **Ejemplo CSS:**

```
.caja-auto {  
  width: 200px;  
  height: 100px;  
  border: 1px solid orange;  
  overflow: auto;  
}
```

- **Resultado:** Si el contenido cabe, se ve como `visible` (pero recortado si fuera necesario, aunque sin barras). Si el contenido desborda verticalmente, aparecerá una barra de desplazamiento vertical. Si desborda horizontalmente (p. ej., una palabra muy larga sin saltos), aparecerá una barra horizontal.

### 5. `clip` :

- Similar a `hidden`, el contenido que desborda se recorta en los límites de la caja (específicamente, en el borde del padding).
- La diferencia clave con `hidden` es que `clip` **prohíbe cualquier tipo de desplazamiento**, incluido el desplazamiento programático (por ejemplo, usando JavaScript `element.scrollTo()`).

- También puede ofrecer ligeras ventajas de rendimiento en algunos navegadores porque indica que no hay necesidad de preparar mecanismos de scroll.
- Es un valor más reciente que los anteriores.
- **Ejemplo CSS:**

```
.caja-clip {
  width: 200px;
  height: 100px;
  border: 1px solid purple;
  overflow: clip;
}
```

- **Resultado:** Funciona como `hidden`, pero con la garantía adicional de que no se puede hacer scroll de ninguna manera.

### Propiedades Específicas por Eje: `overflow-x` y `overflow-y`

También puedes controlar el desbordamiento de forma independiente para el eje horizontal (`overflow-x`) y el eje vertical (`overflow-y`). Puedes usar los mismos valores (`visible`, `hidden`, `scroll`, `auto`, `clip`).

- **Ejemplo:** Crear una caja que solo tenga scroll vertical si es necesario, pero oculte cualquier desbordamiento horizontal.

```
.caja-scroll-y {
  width: 300px;
  height: 150px;
  border: 1px solid darkcyan;
  overflow-y: auto; /* Scroll vertical solo si hace falta */
  overflow-x: hidden; /* Ocultar desbordamiento horizontal */
}
```

- **Nota:** Si `overflow` se establece con un solo valor (p.ej., `overflow: scroll;`), aplica ese comportamiento a ambos ejes. Si `overflow` se establece con dos valores (p.ej., `overflow: hidden auto;`), el primer valor aplica a `overflow-x` y el segundo a `overflow-y`. Sin embargo, usar `overflow-x` y `overflow-y` explícitamente suele ser más claro.

### Casos de Uso Comunes:

- Crear áreas de contenido desplazables (como paneles de chat, bloques de código, listas largas en un espacio limitado).
- Evitar que elementos con contenido dinámico o impredecible rompan el layout general de la página (`overflow: hidden` o `overflow: auto`).
- Implementar efectos visuales donde parte de un elemento debe estar oculta.