

23 CSS: margin & padding

Mientras que `width` y `height` definen el tamaño del contenido, y `border` define su marco, `padding` y `margin` controlan el espacio *alrededor* del contenido y *entre* los elementos, respectivamente. Son fundamentales para crear diseños legibles y bien espaciados.

`padding` (Relleno)

- **¿Qué es?** El `padding` es el espacio **interno** de la caja, situado entre el **área de contenido** y el **borde**.
- **Propósito:** Crear espacio entre el contenido de un elemento y su propio borde. Es como el "acolchado" interior de la caja.
- **Características:**
 - El color o imagen de fondo (`background`) del elemento **se extiende** a través del área de padding.
 - El padding **incrementa** el tamaño total ocupado por la caja si se usa `box-sizing: content-box;` (el valor por defecto).
 - Con `box-sizing: border-box;`, el padding se aplica *dentro* del `width` y `height` definidos, reduciendo el espacio disponible para el contenido.
 - **No es heredable.**
- **Propiedades:**
 - **Shorthand:** `padding: <valor(es)>;`
 - **Longhand:**
 - `padding-top:`
 - `padding-right:`
 - `padding-bottom:`
 - `padding-left:`
- **Sintaxis Shorthand (igual que para `border-width`):**
 - `padding: 10px;` (10px en los 4 lados)
 - `padding: 10px 20px;` (10px arriba/abajo, 20px izquierda/derecha)
 - `padding: 10px 20px 5px;` (10px arriba, 20px izquierda/derecha, 5px abajo)
 - `padding: 10px 20px 5px 15px;` (10px arriba, 20px derecha, 5px abajo, 15px izquierda - en sentido horario)
- **Valores:** Generalmente `<length>` (`px`, `em`, `rem`) o `<percentage>` (%). Los porcentajes en `padding` se calculan **siempre** en relación al **ancho (`width`) del bloque contenedor**, incluso para `padding-top` y `padding-bottom`.
- **Ejemplo:**

```
.boton {  
  background-color: dodgerblue;
```

```

color: white;
border: 1px solid navy;
padding: 10px 25px; /* Añade espacio alrededor del texto del botón */
/* El fondo azul se verá en este espacio de padding */
}

```

margin (Margen)

- **¿Qué es?** El `margin` es el espacio **externo** de la caja, situado **fuera del borde**.
- **Propósito:** Crear espacio **entre** un elemento y otros elementos adyacentes. Es como el "espacio vital" alrededor de la caja.
- **Características:**
 - El margen es **siempre transparente**. El color o imagen de fondo del elemento **no se extiende** al área de margen.
 - Los márgenes verticales (`margin-top` , `margin-bottom`) de elementos de bloque adyacentes pueden **colapsar** (combinarse en un solo margen mayor). Abordaremos esto en detalle más adelante.
 - Los márgenes horizontales (`margin-left` , `margin-right`) **no colapsan**, simplemente se suman.
 - Los márgenes verticales generalmente **no tienen efecto** en elementos en línea no reemplazados (como `` , `<a>`). Los márgenes horizontales sí funcionan.
 - **No es heredable.**
- **Propiedades:**
 - **Shorthand:** `margin: <valor(es)>;`
 - **Longhand:**
 - `margin-top:`
 - `margin-right:`
 - `margin-bottom:`
 - `margin-left:`
- **Sintaxis Shorthand (igual que para `padding`):**
 - `margin: 15px;` (15px en los 4 lados)
 - `margin: 10px 20px;` (10px arriba/abajo, 20px izquierda/derecha)
 - `margin: 10px 20px 30px;` (10px arriba, 20px izquierda/derecha, 30px abajo)
 - `margin: 10px 20px 30px 40px;` (10px arriba, 20px derecha, 30px abajo, 40px izquierda - sentido horario)
- **Valores:**
 - `<length>` (`px` , `em` , `rem`)
 - `<percentage>` (%) - Calculado en relación al **ancho (`width`) del bloque contenedor**.
 - **auto** : Un valor especial. El navegador calcula el margen. Su uso más importante es para **centrar horizontalmente elementos de bloque**: si un elemento de bloque tiene un `width`

definido (no `auto`), y se le asigna `margin-left: auto;` y `margin-right: auto;` (o el shorthand `margin: 0 auto;` o `margin: auto;`), el navegador distribuirá el espacio horizontal sobrante equitativamente entre los márgenes izquierdo y derecho, centrando la caja.

- **Valores Negativos:** A diferencia de `padding`, `margin` sí acepta valores negativos. Un margen negativo puede hacer que un elemento se **solape** con otros elementos o se "acerque" más a ellos. (Veremos más sobre márgenes negativos en una sección posterior).

- **Ejemplo:**

```
h1 {
  margin-bottom: 20px; /* Espacio debajo del H1 */
}
p {
  margin-bottom: 15px; /* Espacio debajo de cada párrafo */
}
.contenedor-principal {
  width: 800px;
  margin: 30px auto; /* 30px arriba/abajo, centrado horizontalmente */
  padding: 20px;
  border: 1px solid #ccc;
}
.error {
  margin-top: -10px; /* Sube 10px, posiblemente solapando el elemento anterior */
}
```

Diferencias Clave: `padding` vs. `margin`

Característica	<code>padding</code> (Relleno)	<code>margin</code> (Margen)
Ubicación	Dentro del borde, alrededor del contenido	Fuera del borde, entre elementos
Fondo	El fondo del elemento se extiende	Siempre transparente
Colapso	No colapsa	Márgenes verticales pueden colapsar
Negativos	No permite valores negativos	Permite valores negativos (para solapamiento)
Propósito	Espacio interior del elemento	Espacio exterior, separación entre elementos
Box-Sizing	Afectado por <code>box-sizing</code>	No afectado directamente por <code>box-sizing</code>

Uso en Reseteos:

Como mencionamos en "Reseteo y Normalización", es muy común que las hojas de estilo de reseteo establezcan `margin: 0;` y `padding: 0;` para la mayoría o todos los elementos (`*`) para eliminar las diferencias predeterminadas de los navegadores y empezar desde una base cero consistente.