Box Model (margin, padding, border)

Every element is a box of 4 layers: content, padding, border, margin — the real width math, shorthand order, border-box and margin collapse.

Introduction

Here is the secret at the heart of CSS layout: **every element is a box**. A heading, a paragraph, an image — each one is a rectangle made of four layers, like a framed photo: the photo itself (**content**), the white mat around it (**padding**), the frame (**border**), and the gap between this frame and the next one on the wall (**margin**). Once you see the four layers, spacing stops being guesswork — and one famous formula explains why your "300px box" measures 344px on screen. *(Roman Urdu: Har element aik dabba hai — andar content, phir padding, phir border, sab se bahar margin.)*

Key points

- **Content** — the text or image itself. `width` and `height` set the size of **this layer only** — that sentence is the key to the whole lesson.
- **Padding** — space *inside* the box, between content and border. Padding takes the box's background colour, so it looks like part of the element. `padding: 20px` = 20px on all four sides.
- **Border** — the visible edge: `border: 2px solid navy` (width, style, colour — solid, dashed, dotted…). Borders count toward the box's real size.
- **Margin** — space *outside* the box, pushing neighbours away. Margin is transparent — the page's background shows through it. Margins are how boxes keep their distance.

  (Roman Urdu: Padding dabbe ke andar ki jagah hai, margin dabbe ke bahar ki doori hai.)
- **The width math (default `content-box`):** total width = width + left/right padding + left/right border. A box with `width: 300px; padding: 20px; border: 2px solid;` is **300 + 40 + 4 = 344px** wide on screen. It surprised every CSS learner in history — once.
- **`box-sizing: border-box` — the fix professionals use:** now `width` means the *total* box, and padding/border are squeezed **inside** it. `width: 300px` stays 300px, whatever the padding. Most projects put `* { box-sizing: border-box; }` at the top of the CSS.
- **Shorthand clock order:** `margin: 10px 20px 30px 40px;` reads **top, right, bottom, left** — clockwise from the top. Two values = vertical, horizontal (`margin: 10px 20px`). One value = all sides.
- **Margin collapse:** two vertical margins that touch **merge into one** — the bigger wins. A 30px bottom margin under a box and a 20px top margin above the next give 30px of gap, not 50px. Padding never collapses.
- **Centring a block:** `margin: 0 auto;` on a block with a set width centres it horizontally — `auto` splits the leftover space equally on both sides.

Code examples

Example 1 — The 344px surprise

```
.box {
  width: 300px;
  padding: 20px;
  border: 2px solid #0b6e4f;
  background: #eafaf1;
}
```

What it does: A content width of 300px, 20px of padding on each side, and a 2px border on each side. DevTools' box diagram (inspect the element — the little layered rectangle at the bottom of the Styles panel) shows all four layers with these exact numbers.

What you will see: measure the box on screen: **344px** — 300 content + 20 + 20 padding + 2 + 2 border. The background colour fills content *and* padding (they are one visible area), the green border wraps them, and the diagram in DevTools matches your rule layer by layer.

Example 2 — Shorthand and centring

```
.card-a { margin: 24px; }              /* all sides 24 */
.card-b { margin: 12px 24px; }         /* top/bottom 12, left/right 24 */
.card-c { margin: 10px 20px 30px 40px; } /* top right bottom left */

.page {
  width: 800px;
  margin: 0 auto;   /* centre the whole column */
}
```

What it does: Three margin shorthands — 1, 2 and 4 values — showing how the missing sides copy their opposite. Then the classic page column: a fixed-width block with `margin: 0 auto`, the two `auto`s splitting the empty space equally.

What you will see: cards with visibly different breathing room (card-c leans left, pushed by its 40px left margin), and an 800px column sitting exactly in the middle of a wide screen. Remove the width from `.page` and the centring vanishes — `auto` needs leftover space to split.

Example 3 — border-box, the professional default

```
* { box-sizing: border-box; }

.half {
  width: 50%;
  padding: 20px;
  border: 2px solid #333;
  float: left;
}
```

What it does: With `border-box`, two `.half` boxes of `width: 50%` sit side by side — each is *exactly* half the row, padding and border included in the 50%. (Under the old `content-box`, each "half" was 50% + 44px, and the second box dropped to the next line — the classic layout bug of the float era, and it still bites in grid and flex maths today.)

What you will see: two equal boxes sharing one row neatly. This one line — `* { box-sizing: border-box; }` — at the top of your stylesheet is the single highest-value line in this lesson.

Quick recap

- Four layers, inside → out: **content** → **padding** → **border** → **margin**.
- Default math: total = width + padding + border (300 + 40 + 4 = **344**).
- **`box-sizing: border-box`** makes width mean the whole box — use it everywhere.
- Shorthand order is a **clock**: top, right, bottom, left. Vertical margins **collapse**; `margin: 0 auto` centres a sized block.

Common mistakes

1. **"I set 300px, why is it 344?"** — you set the *content* to 300px. Padding and border add on top under `content-box`. Either accept the math or switch to `border-box` (recommended).
2. **Expecting margins to add up** — a 30px margin meeting a 20px margin gives 30px of space, not 50. That is margin collapse, and it is by design. Use padding on a parent when you need guaranteed space inside.
3. **Putting the background on the wrong layer in your head** — background covers content + padding, never margin. A "gap with the wrong colour" between boxes is margin showing the *page* background through.

Practice

1. Create a box with `width: 300px; padding: 20px; border: 2px solid` and confirm 344px in DevTools. Then add `box-sizing: border-box` to it and confirm it becomes exactly 300px.

2. Build a centred page column (`width: 720px; margin: 0 auto`) holding three cards; give the cards margins using the 4-value clock shorthand so each card has different spacing — then swap to 2-value shorthand and observe which sides copy which.

Practice quiz

1. What does margin do?

- Space OUTSIDE the box, between it and neighbours
- Space INSIDE the box
- The visible edge
- The text itself

Answer: Space OUTSIDE the box, between it and neighbours

2. What does padding do?

- Space OUTSIDE the box
- Space INSIDE the box, between content and border
- The border width
- The font size

Answer: Space INSIDE the box, between content and border

3. What does border do?

- The space outside
- The visible edge around the box
- The text style
- The background image

Answer: The visible edge around the box

Watch the video lesson: https://youtu.be/JgLnUWgcCoo

Read online: https://scopetreasure.com/notes/css-box-model · More free tools and notes at https://scopetreasure.com