ScopeTreasure Courses

CSS COURSES · LESSON 2.11 (SHORT) · CODE SCOPE SERIES

CSS in 60 Seconds — Hover Effect

One property trick in three lines: a button that lifts on hover — base style, transition, and the :hover rule. From the 60-seconds Shorts.

Introduction

This is the first of the **CSS in 60 Seconds** Shorts: one property trick, fully working, inside a minute. Today's trick is the one you will use more than any other — the **hover lift**. Three short rules: a normal button, a `transition`, and a `:hover` state that raises the button with a shadow. It is Lesson 2.10's transition engine in miniature — and the exact code behind half the buttons on the modern web. *(Roman Urdu: Teen lines, aik professional button — yehi is Short ka vaada hai.)*

Key points

- **`:hover` is a pseudo-class** — a temporary *state* of an element, written with a colon: `button:hover`. It applies while the mouse is over the element and removes itself when the mouse leaves. No JavaScript anywhere.
- **The recipe has an order:** (1) style the base element, (2) put `transition: all 0.3s` on the **base**, (3) describe only the *changes* in `:hover`. The transition is what turns the change from a jump into a glide.
  (Roman Urdu: Pehle button, phir transition, phir hover ki tabdeeli — tartib yehi hai.)
- **`transform: translateY(-3px)`** moves the button up 3px *visually* without disturbing the layout around it — transforms never push neighbours (unlike changing margin or top).
- **A shadow sells the lift:** `box-shadow: 0 8px 16px rgba(0,0,0,0.2)` under a raised button mimics a real object catching light — your eye reads "height" instantly.
- **Touch screens have no hover** — phones tap. Hover is *decoration*, never the only way to reach information; anything important must also work by tap.
- The same three-line pattern works on **cards, links, images, menu items** — learn it once, apply it everywhere.

Code examples

Example 1 — The 60-seconds button

```
button {
  background: #0b6e4f; color: white;
  padding: 12px 24px; border: none; border-radius: 10px;
  transition: all 0.3s ease;
}
button:hover {
  transform: translateY(-3px);
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
}
```

What it does: Line by line in the spirit of the Short: the base rule makes a plain green button and arms the transition; the `:hover` rule changes exactly two things — position (up 3px) and shadow. Because `all` is transitioned, both changes animate together over 0.3s.

What you will see: rest your mouse on the button — it rises gently and a soft shadow blooms beneath it; move away and it settles back. That tiny rise is the difference between a button that feels *dead* and one that feels *pressable*.

Example 2 — The same button WITHOUT the transition

```
button { background: #0b6e4f; color: white; padding: 12px 24px; }
button:hover { transform: translateY(-3px); }  /* no transition anywhere */
```

What it does: Identical hover change, but nothing tells the browser to animate between the states.

What you will see: the button **teleports** 3px on hover and snaps back on leave — jumpy and cheap. Keep this version in mind whenever a hover feels "rough": the missing ingredient is never the `:hover` rule; it is the `transition` on the base.

Example 3 — The trick grows up: cards and links

```
.card { transition: all 0.3s ease; border-radius: 14px; }
.card:hover { transform: translateY(-6px); box-shadow: 0 14px 28px rgba(0,0,0,0.15); }

.link { position: relative; }
.link::after {
  content: ""; position: absolute; left: 0; bottom: -3px;
  width: 0; height: 2px; background: #e8490f;
  transition: width 0.3s ease;
}
.link:hover::after { width: 100%; }
```

What it does: The card reuses the lift with bigger numbers (more height, deeper shadow). The link learns a second trick: an `::after` underline that *grows* from 0 to full width on hover — still just base + transition + hover, aimed at a different property.

What you will see: cards that float up as you pass over them, and menu links whose orange underline draws itself in. Two different "animations", one identical pattern — that is why this Short was worth sixty seconds.

Quick recap

- Hover lift recipe: **base style** → **`transition`** **on the base** → **changes in** **`:hover`**.
- `translateY(-3px)` + `box-shadow` = the physical lift illusion; transforms never disturb the layout.
- Hover decorates; it must never hide the only path to something important (phones cannot hover).

Common mistakes

1. **Transition inside the `:hover` rule** — animate-in works, snap-back on leave. The transition belongs on the base element (the lesson of 2.10, worth repeating until it is reflex).
2. **Oversized movement** — `translateY(-20px)` and `scale(1.3)` turn a button into a jumping toy. Professional lifts are 2–6px. If the movement is the first thing you notice, it is too big.
3. **Hover-only content** — a menu or a price that appears *only* on hover is invisible to every phone user. If it matters, it must show (or open) on tap too.

Practice

1. Recreate Example 1 from memory — no peeking — then verify the three ingredients are all present (base, transition, hover changes).
2. Apply the lift to a card, a button and an image on one page, each with slightly different numbers; then add `:focus` next to `:hover` (`button:hover, button:focus { ... }`) so keyboard users get the same feedback.

Practice quiz

1. What is :hover?

- A JavaScript function
- A pseudo-class, a temporary state while the mouse is over the element
- A new HTML tag
- A colour name

Answer: A pseudo-class, a temporary state while the mouse is over the element

2. What is the hover recipe, in order?

- Hover changes first, then the base style

- Base style, then transition on the base, then only the changes in :hover
- Transition inside :hover, then the base style
- Base style, then JavaScript

Answer: Base style, then transition on the base, then only the changes in :hover

3. Why must hover never be the only way to reach something important?

- Because hover is slow
- Because touch screens have no hover, phones tap
- Because hover does not work on buttons
- Because search engines ban hover

Answer: Because touch screens have no hover, phones tap

Watch the video lesson: https://youtube.com/shorts/eyeGdV5gtIU

Read online: https://scopetreasure.com/notes/css-in-60-seconds-hover · More free tools and notes at https://scopetreasure.com