

Going further

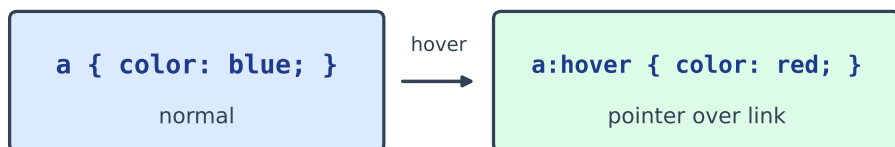
HTML & CSS Reference

Hover & pseudo-classes

A pseudo-class styles an element in one particular state. `:hover` applies while the mouse is over the element. Add `transition` so the change happens smoothly instead of jumping.

```
<style>
  .btn { background: #2563eb; color: white; padding: 10px 18px;
        border: none; border-radius: 8px; font-size: 16px;
        transition: background 0.3s; }
  .btn:hover { background: #16a34a; }
</style>
<button class="btn">Hover me</button>
```

- Other useful pseudo-classes: `:focus` (an input the user clicked into) and `:first-child`.



Exam tip: `:hover`, `:focus`, `:active` are pseudo-classes — write selector:pseudo

`:hover` styles an element while the pointer is over it

Positioning

Positioning moves a box away from its normal place. `relative` nudges the box (and makes it the anchor for its children); `absolute` places a box exactly, measured from the nearest positioned parent; `fixed` pins it to the screen even when the page scrolls.

```
<style>
  .card { position: relative; width: 260px; height: 90px;
        background: #dbeafe; border-radius: 10px; padding: 12px; }
  .badge { position: absolute; top: -10px; right: -10px;
        background: #dc2626; color: white; padding: 4px 10px;
        border-radius: 999px; font-size: 13px; }
</style>
<div class="card">
  A product card
  <span class="badge">NEW</span>
</div>
```

- `top`, `right`, `bottom`, `left` say how far to move — they only work on positioned boxes.

Media queries & responsive design

A responsive page adapts to the screen size. A media query applies its rules only while a condition about the screen holds, such as `max-width`.

```
<style>
  .banner { background: #2563eb; color: white; padding: 16px; }
  @media (max-width: 600px) {
    .banner { background: #16a34a; } /* narrow screens turn green */
  }
</style>
<div class="banner">Resize the window: blue when wide, green when
  ↳ narrow.</div>
```

- Design mobile-first : write the phone styles first, then add `@media (min-width: ...)` rules for bigger screens.
- Percent widths and `max-width` let boxes shrink with the screen.

Common mistakes

- `:hover` sticks to its selector with no space: `.btn:hover`, not `.btn :hover`.

- `position: absolute` is measured from the nearest **positioned** parent — give that parent `position: relative`.
- A media query wraps whole rules: `@media (max-width: 600px) { .a { ... } }`.