ScopeTreasure Courses

CSS COURSES · LESSON 2.1 · CODE SCOPE SERIES

CSS Kya Hai? + 3 Ways to Add CSS

What CSS is, the selector–property–value rule, and the 3 ways to add CSS (inline, internal, external) — plus who wins when they fight: the cascade.

Introduction

In the HTML series you built pages that worked — but looked like plain documents. **CSS (Cascading Style Sheets)** is the design layer of the web: it decides colours, fonts, sizes, spacing and layout. HTML is the skeleton of the page; CSS is its clothes and style. In this lesson you learn what a CSS rule looks like, the three ways to attach CSS to a page, and the "cascade" — the rule that decides which style wins when two styles fight. *(Roman Urdu: HTML page ka dhancha hai, CSS us ki sajawat hai — rang, font, size, jagah, sab CSS tay karta hai.)*

Key points

- A CSS rule has three parts: a **selector** (which element?), a **property** (what to change?) and a **value** (change it to what?) — written `selector { property: value; }`. Example: `p { color: blue; }` means "every paragraph — colour — blue".
  (Roman Urdu: Selector nishana hai, property woh cheez hai jo badalni hai, value us ki nayi shakal hai.)
- One rule can carry **many properties** — separate them with semicolons: `h1 { color: navy; font-size: 32px; text-align: center; }`.
- **Way 1 — Inline:** CSS inside the tag itself with the `style` attribute: `<p style="color: red;">`. Fast for one element, messy for a whole page.
- **Way 2 — Internal:** a `<style>` block inside the page's `<head>`. All the page's styles in one place — good for one-page demos and learning.
- **Way 3 — External:** the styles live in a separate **`.css` file**, and the page links it: `<link rel="stylesheet" href="styles.css">`. This is the professional way — one file styles the whole website, and the browser caches it.
- **The cascade** (the "C" in CSS): when the same element is styled from more than one place, the browser follows a fixed order — **inline beats internal/external**, and between two equal rules the **later** one wins. In Lesson 2.2 you will see the full ladder (specificity).
- The browser reads your HTML first, builds the page tree, then applies the CSS on top — a style can never change the *meaning* of the HTML, only its look.

Code examples

Example 1 — Your first CSS rules (internal)

```
<!DOCTYPE html>
<html>
<head>
  <title>My Profile</title>
  <style>
    body { background-color: #f2f5f9; font-family: Arial, sans-serif; }
    h1 { color: navy; text-align: center; }
    p { color: #333333; font-size: 18px; }
  </style>
</head>
<body>
  <h1>Muhammad Shahid</h1>
  <p>I am learning web development.</p>
</body>
</html>
```

What it does: The `<style>` block in the head holds three rules. `body` sets the page background and the default font for everything. `h1` turns the heading navy and centres it. `p` makes paragraphs dark grey and a little bigger.

What you will see: the plain profile page from the HTML series, transformed — soft grey-blue page, navy centred heading, readable paragraph. Same HTML as before; only the `<style>` block is new. That difference is exactly what CSS is.

Example 2 — The same page, all 3 ways at once

```
<head>
  <link rel="stylesheet" href="styles.css">
  <style>
    .note { color: green; }
  </style>
</head>
<body>
  <p style="color: red;">Inline beats everyone.</p>
  <p class="note">Internal style — green.</p>
  <p>External file styles me.</p>
</body>
```

```
/* styles.css */
p { color: purple; }
```

What it does: Three paragraphs, three styling sources. The first carries an **inline** style. The second matches the **internal** `.note` rule. The third is reached only by the **external** file's `p` rule — which also *reaches* the first two paragraphs, but loses to their stronger sources.

What you will see: paragraph 1 is **red** (inline wins over the external purple), paragraph 2 is **green** (the internal class rule beats the external element rule), paragraph 3 is **purple** (only the external rule applies). One page, three winners — the cascade in action.

Example 3 — The professional setup (external file)

```
<!-- index.html -->
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

```
/* styles.css — the whole site's design lives here */
h1 { color: #0b6e4f; }
.card { background: white; padding: 16px; border-radius: 10px; }
```

What it does: The HTML page only *links* the style file. Every page of the site links the same `styles.css`, so changing one line there re-designs the whole website — and after the first visit the browser keeps the file in cache, so other pages load faster.

What you will see: the page styled exactly as if the CSS were written inside it. The difference is invisible to visitors and enormous for you: one file to edit, zero copy-paste.

Quick recap

- CSS = the design layer: colours, fonts, spacing, layout. A rule is `selector { property: value; }`.
- 3 ways: **inline** (`style=` attribute), **internal** (`<style>` in head), **external** (`.css` file + `<link>`) — external is the professional default.
- The **cascade** decides fights: inline wins; the full ladder comes in Lesson 2.2.

Common mistakes

1. **Forgetting the semicolon or a curly brace** — one missing `;` or `}` can silently kill the whole rule (and sometimes the next one). When a style "does nothing", count your braces first.
2. **Writing CSS in the body without a home** — CSS text pasted between tags shows up as page text. Styles only live in a `style` attribute, a `<style>` block, or a `.css` file.
3. **Styling everything inline** — ten inline styles means ten places to fix when the design changes. Inline is for a one-off override, not a design system.

Practice

1. Take your profile page from the HTML series and style it all three ways: one element inline, the heading rules internal, then move everything into a `styles.css` file and link it. The page must look identical at each step.
2. Using internal CSS, change five things on one page: heading colour, page background, paragraph size, link colour, and centre the heading. Write the five rules from memory.

Practice quiz

1. What does text-align: center do?

- Centres the text inside its box
- Makes text bold
- Moves the box
- Underlines text

Answer: Centres the text inside its box

Watch the video lesson: https://youtu.be/7CFjAJOYd4g

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