CSS — Complete Preparation Textbook (2025)

A practical, beginner-to-advanced handbook. Each topic includes: **What it is → Why it matters → Syntax → Example → Expected behaviour**. Copy the snippets into `index.html` + `styles.css`. Open `index.html` in a browser and observe.

Table of Contents

---

0) Getting Started

Where CSS goes 1. **External file (recommended):** `<link rel="stylesheet" href="styles.css">` 2. **Internal:** `<style> … </style>` inside `<head>` 3. **Inline (avoid):** `<p style="color:red">` — hard to maintain and fights CSP.

Order matters: Later rules override earlier ones (if specificity ties). External → internal `<style>` → inline.

Boilerplate

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>CSS Starter</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Hello CSS</h1>
</body>
</html>
```

---

1) Syntax, Comments, Cascade & Specificity

Syntax

```
selector { property: value; /* comment */ }
```

Comments: `/* … */`

Specificity (strength) - Inline style (1000) > ID `#id` (100) > Class/Attr/Pseudo-class `.c` `[type]` `:hover` (10) > Type `h1` (1) - Ties break by **later** rule. `!important` wins within the same origin/layer (use sparingly).

Example

```
h1 { color: darkslategray; }      /* 1 */
.title { color: teal; }            /* 10 */
#hero .title { color: purple; }    /* 110 */
```

Expected: Purple, because the ID selector is most specific.

---

2) Selectors (from basic to advanced)

Basic

```css
/* Type, class, ID */
main { padding: 1rem; }
.card { border: 1px solid #e5e7eb; }
#login { max-width: 28rem; }
```

Combinators

```css
article h2 { }      /* descendant */
article > h2 { }    /* child */
h2 + p { }          /* adjacent sibling */
h2 ~ p { }          /* general sibling */
```

Attribute

```css
input[type="email"] { border-color: #2563eb; }
a[href^="https://"]::after { content: "↗️"; }
```

Pseudo-classes

```css
a:link{} a:visited{} a:hover{} a:active{}
input:focus-visible{ outline: 3px solid #38bdf8; }
:root:has(dialog[open]) body{ overflow:hidden; }
```

Pseudo-elements

```css
p::first-line{ font-weight:600 }
.button::before{ content:"★"; margin-right:.5em }
```

Level-4 helpers

```css
/* :is(): groups selectors WITHOUT adding specificity */
:is(h1,h2,h3){ margin-block: .75em; }
/* :where(): like :is but zero specificity (great for resets) */
:where(h1,h2,h3,p){ margin: 0; }
/* :has(): parent selector */
.card:has(img){ border-color:#a855f7 }
```

---

3) Inheritance & Value Keywords

Inheritance: Some properties (e.g., `color`, `font-family`) inherit by default; others don't (`margin`, `padding`).

Global keywords: `inherit`, `initial`, `unset`, `revert`.

```
button{ all: unset; /* resets to inherited/initial defaults; use carefully */ }
```

Cascade Layers `@layer`

Organize CSS into layers to control cascade order explicitly.

```
@layer reset, base, components, utilities; /* order */
@layer reset {
  :where(h1,h2,h3,p){ margin:0 }
}
@layer base { body{ font-family:system-ui } }
@layer components { .card{ padding:1rem; } }
@layer utilities { .mt-4{ margin-top:1rem } }
```

---

4) Units

- **Absolute:** `px`, `cm`, `mm`, `in`, `pt` (screen design favors `px`).
- **Font-relative:** `em` (parent), `rem` (root), `ex`, `ch`, `lh`, `rlh`.
- **Viewport:** `vw`, `vh`, `vmin`, `vmax` + modern **dynamic** (`dvh`, `lvh`, `svh`) and logical `vi`/`vb` (inline/block axes).
- **Container:** `cqw`, `cqh`, `cqi`, `cqb` (with container queries).
- **Time/Angles:** `s`, `ms`, `deg`, `rad`, `turn`. **Resolution:** `dpi`, `dppx`.

Example

```
html{ font-size:16px }
h1{ font-size:2rem }         /* 32px */
.card{ width: min(90vw, 60ch) }
```

---

5) Colors

Legacy: `#333`, `rgb(34,197,94)`, `hsl(150 60% 45%) / 0.8` (slash or space/alpha).

Modern: perceptually uniform color spaces and wide gamut.

```
h1{ color: lch(60% 70 30) }                               /* LCH */
.subtitle{ color: lab(45% -5 20) }
.brand{ color: color(display-p3 0.2 0.5 0.9) }
.badge{ color: color-mix(in oklab, #2563eb 60%, white) }  /* mix */
```

---

6) Box Model & Logical Properties

Box model: `content` → `padding` → `border` → `margin`.

```
*{ box-sizing: border-box; }
.card{ padding:16px; border:1px solid #e5e7eb; margin:16px; border-radius: 12px }
```

Logical properties (writing-mode aware):

```
section{ padding-block:1rem; padding-inline:2rem; }
.box{ margin-block-start:1rem; inset-inline: 0; }
```

Overflow

```
.scroll{ overflow:auto; overscroll-behavior:contain }
```

---

7) Display & Flow

```
.inline{ display:inline }
.block{ display:block }
.inline-block{ display:inline-block }
.flow-root{ display:flow-root } /* establishes new block formatting context */
```

Multi-column

```
.columns{ column-count:3; column-gap:1rem; column-rule:1px solid #e5e7eb }
```

---

8) Positioning

```
.rel{ position:relative }
.abs{ position:absolute; top:0; right:0 }
.fixed{ position:fixed; inset:0 0 auto auto } /* top-right */
.sticky{ position:sticky; top:0 }
.z{ z-index:10 }
```

---

9) Float & Clear (legacy)

```
img.float{ float:right; margin:0 0 1rem 1rem }
.clearfix{ display:flow-root } /* modern clearfix */
```

---

10) Flexbox

Container

```
.flex{ display:flex; gap:12px; }
.flex.center{ justify-content:center; align-items:center }
.flex.between{ justify-content:space-between }
```

Item

```
.item{ flex: 1 1 220px }  /* grow, shrink, basis */
.item.auto{ flex: 0 0 auto }
```

Pattern – Navbar

```
.nav{ display:flex; justify-content:space-between; align-items:center }
.links{ display:flex; gap:12px }
```

---

11) Grid

Basic grid

```
.grid{ display:grid; grid-template-columns: repeat(3, 1fr); gap:12px }
.card{ padding:12px; border:1px solid #e5e7eb; border-radius:10px }
```

Auto-fit cards

```
.grid{ grid-template-columns: repeat(auto-fit, minmax(220px, 1fr)); }
```

Areas

```
.layout{ display:grid; grid-template-areas:
  "header header"
  "main   aside"
  "footer footer"; grid-template-columns:2fr 1fr; gap:12px }
```

```
header{ grid-area:header } main{ grid-area:main } aside{ grid-area:aside }
footer{ grid-area:footer }
```

12) Sizing, Aspect Ratio, Overflow

```
img{ max-width:100%; height:auto; }
.card{ min-height:200px; max-width:60ch }
.thumb{ aspect-ratio: 16 / 9; object-fit: cover; }
.scroll{ overflow: auto }
```

13) Typography

Fonts & Variable Fonts

```
@font-face{
  font-family: 'InterVar';
  src: url('/fonts/Inter-Variable.woff2') format('woff2');
  font-weight: 100 900; /* variable */
  font-display: swap;
}
body{ font-family: InterVar, system-ui, Arial, sans-serif; line-height:1.6 }
h1{ font-weight: 700; font-stretch: 100%; }
```

Features & Spacing

```
p{ hyphens:auto; text-wrap:balance; }
code{ font-family: ui-monospace, SFMono-Regular, Menlo, Consolas, monospace }
.lead{ font-size: clamp(1.125rem, 1rem + 1vw, 1.5rem) }
.smallcaps{ font-variant-caps: small-caps }
.numtab{ font-variant-numeric: tabular-nums }
```

Text decoration & overflow

```
.link{ text-decoration: underline; text-underline-offset: 2px }
.ellipsis{ white-space:nowrap; overflow:hidden; text-overflow:ellipsis; max-width:30ch; display:block }
```

14) Lists & Generated Content

```
ul{ list-style: square inside }
li::marker{ color:#2563eb }
/* Counters */
ol{ counter-reset: step }
ol > li{ counter-increment: step }
ol > li::marker{ content: counters(step, '.') '. ' }
/* Before/after */
.btn::before{ content:'→ '; }
```

15) Backgrounds, Borders, Outline, Shadows

```
.box{ background:#f8fafc url(bg.svg) no-repeat right 1rem top 1rem / 80px }
.box{ border:1px solid #e5e7eb; border-radius:12px }
.box{ outline:2px dashed #94a3b8; outline-offset:4px }
.shadow{ box-shadow: 0 8px 24px rgba(0,0,0,.12) }
```

Multiple backgrounds: `background: url(a.svg), linear-gradient(#fff, #f3f4f6);`

16) Filters, Blend & Backdrop

```
.photo{ filter: grayscale(.6) contrast(1.1) }
.overlay{ mix-blend-mode: multiply }
.blurglass{ backdrop-filter: blur(8px); background: rgba(255,255,255,.4) }
```

17) Shapes & Masks

```
.avatar{ clip-path: circle(45%) }
.badge{ clip-path: polygon(0 0, 100% 0, 100% 60%, 0 100%) }
.mask{ -webkit-mask-image: radial-gradient(circle, #000 60%, transparent
61%); mask-image: radial-gradient(circle, #000 60%, transparent 61%); }
```

18) Transforms (2D/3D)

```
.card{ transform: translateY(-2px) scale(1.02); transform-origin: center; }
.cube{ transform: rotateX(20deg) rotateY(30deg); perspective: 800px }
```

19) Transitions

```
.button{ transition: transform .2s ease, box-shadow .2s }
.button:hover{ transform: translateY(-2px); box-shadow: 0 8px 24px rgba(0,0,
0,.15) }
/* Custom timing */
.soft{ transition-timing-function: cubic-bezier(.2,.6,.2,1) }
```

20) Animations

```
@keyframes pulse { from{ opacity:.6 } to{ opacity:1 } }
.badge{ animation: pulse .8s infinite alternate }
@keyframes bounce { 0%,100%{ transform:translateY(0) } 50%{
transform:translateY(-8px) } }
.dot{ animation: bounce 1s infinite; animation-timing-function:
steps(2,end) }
```

Performance: Prefer transform/opacity; avoid animating layout (width, top) where possible.

21) Scroll Behaviour & Snap

```
html{ scroll-behavior: smooth }
.carousel{ scroll-snap-type: x mandatory }
.slide{ scroll-snap-align: start }
/* Offset anchor jumps */
:target{ scroll-margin-top: 6rem }
```

22) Media Queries (Responsive & Preferences)

```
/* Breakpoints */
@media (min-width: 640px){ .grid{ grid-template-columns: repeat(2,1fr) } }
@media (min-width: 1024px){ .grid{ grid-template-columns: repeat(3,1fr) } }
/* Dark mode */
@media (prefers-color-scheme: dark){ :root{ --bg:#0f172a; --fg:#e2e8f0 }
body{ background:var(--bg); color:var(--fg) } }
/* Reduced motion */
@media (prefers-reduced-motion: reduce){ *{ animation:none!important;
transition:none!important } }
```

```
/* Hover capability */
@media (hover: hover){ .button:hover{ transform:translateY(-2px) } }
```

23) Container Queries (Component-level responsiveness)

```
<!-- HTML: mark a container -->
<section class="gallery">
  <div class="card">…</div>
  <div class="card">…</div>
</section>
```

```
.gallery{ container-type: inline-size; /* container for queries */
display:grid; gap:12px }
.card{ background:#fff; padding:1rem; border:1px solid #e5e7eb }
@container (min-width: 480px){ .gallery{ grid-template-columns: repeat(2,
1fr) } }
@container (min-width: 800px){ .gallery{ grid-template-columns: repeat(3,
1fr) } }
/* Container units */
.card{ font-size: clamp(1rem, 2cqw, 1.25rem) }
```

24) Feature Queries `@supports`

```
@supports (backdrop-filter: blur(4px)){
  .panel{ backdrop-filter: blur(4px) }
}
@supports not (aspect-ratio: 1){ .thumb{ height:0; padding-top:56.25% } }
```

25) Printing `@page` & Print Styles

```
@media print{
  nav, .no-print{ display:none }
  a::after{ content:" (" attr(href) ")" }
}
@page{ margin: 12mm }
```

26) Forms Styling

```css
input, select, textarea{ font: inherit; color:inherit }
input::placeholder{ color:#9ca3af }
input:focus-visible{ outline:3px solid #38bdf8; outline-offset:2px }
/* One-line theming */
:root{ accent-color:#2563eb }
/* Group focus */
.field:focus-within{ box-shadow:0 0 0 3px rgba(56,189,248,.35) }
```

---

27) Accessibility-Friendly CSS

- Always style **focus** (`:focus-visible`), maintain contrast.
- Respect user preferences: `prefers-reduced-motion`, `prefers-contrast`.
- Avoid conveying meaning by color alone; add icons/text.
- Hit targets: `min-height: 44px` for tap areas.

---

28) Cascade Management with `@layer`

Pattern:

```css
@layer reset, base, components, utilities;
@layer reset { *,*::before,*::after{ box-sizing:border-box } }
@layer base { body{ margin:0; font: 16px/1.6 system-ui } }
@layer components { .card{ padding:1rem; border:1px solid #e5e7eb; border-radius:12px } }
@layer utilities { .mt-4{ margin-top:1rem } .text-center{ text-align:center } }
```

---

29) Performance & Debugging

- Audit with **Lighthouse**; check **Coverage** in DevTools to spot unused CSS.
- Avoid huge selector chains; prefer component scopes.
- Animate transform/opacity only.
- Use `content-visibility: auto` for large off-screen sections.

```css
.section{ content-visibility:auto; contain-intrinsic-size: 800px 600px }
```

---

30) Mini-Reference (Cheat-Sheet)

Common Properties by Area

Layout & Box: `display`, `position`, `top/right/bottom/left`, `inset`, `z-index`, `float`, `clear`, `box-sizing`, `width/height`, `min/max-*`, `margin`, `padding`, `border*`, `overflow*`, `aspect-ratio`

Flex: `display:flex`, `flex-direction`, `flex-wrap`, `justify-content`, `align-items`, `align-content`, `gap`, `row-gap`, `column-gap`, `flex`, `flex-grow`, `flex-shrink`, `flex-basis`, `order`, `align-self`

Grid: `display:grid`, `grid-template-columns/rows/areas`, `grid-auto-flow`, `grid-auto-columns/rows`, `grid-column/row`, `gap`

Typography: `font-*`, `@font-face`, `line-height`, `letter-spacing`, `word-spacing`, `text-align`, `text-wrap`, `text-transform`, `text-decoration-*`, `text-shadow`, `hyphens`

Visual: `color`, `background*`, `box-shadow`, `filter`, `backdrop-filter`, `opacity`, `mix-blend-mode`

Effects: `transform*`, `transition*`, `animation*`, `will-change`

A11y & Interactions: `:focus-visible`, `:focus-within`, `cursor`, `pointer-events`, `accent-color`, `caret-color`, `scroll-behavior`, `scroll-snap-*`

Theming: `--custom-prop`, `var()`, `@layer`, `@supports`, `@media`, `@container`, `color-mix()`

Full Examples

A) Responsive Card Grid (Grid + container queries)

```html
<main class="catalog">
  <article class="card"><h3>Item A</h3><p>Details</p></article>
  <article class="card"><h3>Item B</h3><p>Details</p></article>
  <article class="card"><h3>Item C</h3><p>Details</p></article>
</main>
```

```css
.catalog{ container-type:inline-size; display:grid; gap:12px }
.card{ padding:1rem; border:1px solid #e5e7eb; border-radius:12px }
@container (min-width:480px){ .catalog{ grid-template-columns:repeat(2, 1fr) } }
@container (min-width:820px){ .catalog{ grid-template-columns:repeat(3, 1fr) } }
```

Expected: Cards flow into 1/2/3 columns based on the container width, not the viewport.

B) Navbar (Flex) with Focus Styles

```
<header class="nav">
  <a class="logo" href="#">Brand</a>
  <nav class="links">
    <a href="#">Home</a>
    <a href="#">About</a>
    <a href="#">Contact</a>
  </nav>
</header>
```

```
.nav{ display:flex; justify-content:space-between; align-items:center; padding:12px 16px; border-bottom:1px solid #e5e7eb }
.links{ display:flex; gap:12px }
.links a{ padding:6px 10px; border-radius:8px }
.links a:focus-visible{ outline:3px solid #38bdf8; outline-offset:2px }
```

Expected: Horizontal bar with accessible focus outlines on links.

C) Theme with Custom Properties + Dark Mode

```
@layer base {
  :root{ --bg:#ffffff; --fg:#111827; --brand:#2563eb }
  @media (prefers-color-scheme: dark){ :root{ --bg:#0f172a; --fg:#e2e8f0; --brand:#38bdf8 } }
  body{ background:var(--bg); color:var(--fg) }
  .btn{ background:var(--brand); color:#fff; padding:.5rem 1rem; border-radius:8px }
}
```

Expected: Auto dark theme based on user preference.

End of CSS Preparation Textbook (2025)