

Coding Konvention für CSS

1. Struktur und Organisation

- Nutze Kommentare zur Trennung und Dokumentation der Abschnitte
- Nutze eine einheitliche Reihenfolge von Eigenschaften (z. B. Position, Box-Modell, Typografie, Farbe, Sonstiges)
- Gruppiere verwandte Selektoren sinnvoll zusammen

2. Methodik und Namenskonvention

- Nutze **sprechende Klassennamen** (englisch, keine Abkürzungen wie `.bx` oder `.clr`)
- Vermeide zu generische Klassen wie `.red`, `.big`, `.container`
- Keine IDs für Styling, nur für JS oder interne Verlinkung

3. Lesbarkeit und Wartbarkeit

- Verwende konsistente Einrückung (z. B. 2 oder 4 Leerzeichen)
- Schreibe jede CSS-Regel in eine neue Zeile
- Nutze Leerzeilen zwischen Regelblöcken
- Halte den Code kurz, klar und modular

```
css
.active_button {
  display: inline-block;
  padding: 0.5rem 1rem;
  background-color: var(--primary);
  color: #fff;
  border-radius: 4px;
}
```

4. Responsives Design

- Cluster Medienabfragen (@media) bspw. unten im Code
- Flexbox und Grid statt floats und positioning
- Es ist dir überlassen, ob du mobile-first oder desktop-first arbeitest!
- Begrenze den Content für große Auflösungen. Das Design hat die volle Breite!



Werde KI Softwareentwickler nach deinem Zeitplan!

Lerne Programmieren, beherrsche neueste KI-Anwendungen und starte anschließend deinen ersten Job in der IT-Branche.



[Jetzt bewerben](#)

[Kostenlose Beratung](#)

Zertifiziert und geschätzt von:

Hervorragend
★★★★★
258 Bewertungen auf
Trustpilot



5. Best Practices

- Verwende **CSS Custom Properties (Variablen)**:

```
css
:root {
    --primary-color: #0066cc;
    --spacing-base: 1rem;
}

.button {
    padding: var(--spacing-base);
    background: var(--primary-color);
}
```

- Vermeide **!important**, außer bei Edge-Cases
- Entferne ungenutztes CSS regelmäßig

6. Quellen

- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- <https://css-tricks.com/>
- <https://getbem.com/introduction/>
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>