

# Tag 1 – Einführung in **CSS (Cascading Style Sheets)**

---

## 1. Was ist CSS?

CSS steht für **Cascading Style Sheets** und ist eine Stylesheet-Sprache, mit der das **Aussehen und Layout von HTML-Dokumenten** gestaltet wird. Während HTML die Struktur und den Inhalt einer Webseite definiert, ist CSS für deren visuelle Gestaltung zuständig.

Ziel: Trennung von Inhalt (HTML) und Präsentation (CSS)

Vorteile von CSS:

- **Gestaltung:** Schriftarten, Farben, Abstände, Positionierung, Animationen u.v.m.
- **Wiederverwendbarkeit:** Ein Stylesheet kann für mehrere HTML-Dateien verwendet werden.
- **Skalierbarkeit:** Große Projekte können modular gestylt werden.
- **Responsivität:** CSS ermöglicht dynamisches, geräteübergreifendes Design.
- **Barrierefreiheit:** CSS kann zur Verbesserung der Zugänglichkeit beitragen.

---

## 2. CSS-Syntax

```
selector {  
  property: value;  
}
```

Bestandteile:

- **Selektor:** Welches HTML-Element wird angesprochen?  
z.B. `h1`, `p`, `.button`, `#header`
- **Eigenschaft (property):** Was soll verändert werden?  
z.B. `color`, `margin`, `font-size`, `background-color`
- **Wert (value):** Wie genau soll es geändert werden?  
z.B. `blue`, `16px`, `auto`

Beispiel:

```
p {  
  color: green;  
  font-size: 18px;  
}
```

Alle `<p>`-Elemente werden **grün** dargestellt und haben eine **Schriftgröße von 18 Pixeln**.

---

## 3. CSS Einbindungsmöglichkeiten

### 1. Inline-CSS

Direkt am HTML-Element:

```
<h1 style="color: red;">Achtung!</h1>
```

**Vorteil:** Schnell, gezielt

**Nachteil:** Schwer wartbar, nicht wiederverwendbar

---

## 2. Internes CSS (im `<head>`)

```
<head>
  <style>
    p {
      font-style: italic;
    }
  </style>
</head>
```

**Vorteil:** Alles in einer Datei

**Nachteil:** Gilt nur für *eine* Seite, unübersichtlich bei komplexem Layout

---

## 3. Externes CSS (empfohlene Methode)

```
<link rel="stylesheet" href="styles.css">
```

```
/* styles.css */
body {
  background-color: #fafafa;
}
```

**Vorteil:** Saubere Trennung von Struktur und Stil, wiederverwendbar

**Nachteil:** Eine zusätzliche Datei muss geladen werden

---

## 4. Selektoren in CSS (Einleitung)

Einfache Selektoren:

| Typ               | Beispiel             | Bedeutung                                  |
|-------------------|----------------------|--|
| Element-Selektor  | <code>p</code>       | alle <code>&lt;p&gt;</code> -Elemente      |
| Klassenselektor   | <code>.box</code>    | alle Elemente mit <code>class="box"</code> |
| ID-Selektor       | <code>#header</code> | das Element mit <code>id="header"</code>   |
| Universalselektor | <code>*</code>       | alle HTML-Elemente                         |

**Hinweis:** IDs sollten einzigartig sein, Klassen können mehrfach verwendet werden.

---

## 5. Vererbung & Kaskadierung

Vererbung:

Bestimmte CSS-Eigenschaften (z. B. `color`, `font-family`) **vererben sich** an untergeordnete Elemente.

Beispiel:

```
body {
  color: blue;
}
```

Alle Texte innerhalb des `<body>`-Elements erscheinen in Blau – sofern keine anderen Farben angegeben sind.

## Kaskadierung:

Wenn mehrere Regeln auf dasselbe Element zutreffen, entscheidet die **Kaskade** (Reihenfolge, Spezifität, Herkunft), welche Regel gilt.

Beispiel:

```
p {  
  color: red;  
}  
#text {  
  color: green;  
}
```

Wenn `<p id="text">Hallo</p>` im HTML steht, ist die Farbe **grün**, weil der ID-Selektor spezifischer ist.

---

## 6. Beispiel: HTML + externe CSS-Datei

index.html

```
<!DOCTYPE html>  
<html lang="de">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>CSS Demo</title>  
  <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
  <h1>Willkommen!</h1>  
  <p>Dies ist ein Absatz mit CSS-Stil.</p>  
</body>  
</html>
```

styles.css

```
body {  
  background-color: #f4f4f4;  
  font-family: Arial, sans-serif;  
}  
  
h1 {  
  color: darkblue;  
  text-align: center;  
  font-size: 32px;  
}  
  
p {  
  color: #666;  
  font-size: 18px;  
}
```

---

## 7. Übungsaufgaben: CSS-Grundlagen

### Aufgabe 1:

Erstelle eine HTML-Seite mit einem Absatz und einer Überschrift. Wende Inline-CSS an, um den Text rot und fett darzustellen.

---

### Aufgabe 2:

Verwende **internes CSS**: Style ein `<h2>`-Element mit blauer Farbe und `text-align: center`.

---

### Aufgabe 3:

Erstelle eine HTML-Datei `index.html` und eine separate CSS-Datei `style.css`. Verlinke beide korrekt. Style den Hintergrund der Seite in Hellgrau, Überschriften in Dunkelgrün und den Absatztext in Grau.

---

### Aufgabe 4:

Verwende Klassenselektoren. Erstelle drei `<div>`-Boxen mit der Klasse `.box`. Gib ihnen unterschiedliche Farben, aber gleiche Größe (100px x 100px). Zeige sie nebeneinander an.

---

### Aufgabe 5:

Nutze einen ID-Selektor, um ein bestimmtes Element (`#highlight`) mit gelbem Hintergrund hervorzuheben. Verwende innerhalb desselben Elements auch eine Klasse, z. B. `.bold`, um den Text fett zu machen.

---

### Aufgabe 6:

Was passiert, wenn man `color: red;` auf das `<body>`-Element anwendet und gleichzeitig `color: green;` auf ein `<p>` innerhalb des Body? Teste es aus und erkläre.

---

### Aufgabe 7:

Erstelle eine eigene Mini-Webseite mit externem CSS, die Folgendes enthält:

- Eine Überschrift
- Zwei Absätze
- Mindestens zwei verschiedene Farben
- Verwendung von mindestens einer Klasse und einer ID