

# Modul 293 - Aufgabenblatt 03

von Lukas Meier

Unterricht vom 09.12.2025

## 1 Aufgaben

In dieser Lektion vertiefen Sie Ihr Wissen über das CSS Box-Modell sowie das Layout-Verhalten von Inline- und Block-Elementen. Arbeiten Sie die folgenden Aufgaben der Reihe nach durch und beobachten Sie die Auswirkungen Ihrer Änderungen jeweils im Browser.

### 1.1 Vorbereitung

Öffnen Sie Ihr bestehendes Projekt aus den vorherigen Lektionen (index.html, contact.html sowie styles.css). Stellen Sie sicher, dass Ihre CSS-Datei korrekt im `head`-Bereich der HTML-Dateien eingebunden ist.

### 1.2 Ein einfaches Box-Element

Erstellen Sie in der Datei index.html unterhalb des Headers ein neues `div`-Element. Vergeben Sie diesem Element die Klasse `box`. Befüllen Sie das `div` mit einem kurzen Text Ihrer Wahl.

Erstellen Sie in der Datei styles.css eine CSS-Regel für die Klasse `box` und definieren Sie folgende Eigenschaften:

- Breite: 300px
- Hintergrundfarbe: hellgrau

Laden Sie die Seite im Browser neu. Wie verhält sich das Element im Layout?

## 1.3 Padding und Border

Erweitern Sie die bestehende CSS-Regel der Klasse `box` um folgende Eigenschaften:

- Padding: 20px
- Border: 2px solid schwarz

Beobachten Sie, wie sich der Abstand zwischen Text und Rahmen verändert. Welche Rolle spielt das Padding?

## 1.4 Margin

Ergänzen Sie der `box`-Klasse nun noch eine Margin von 30px.

Beobachten Sie den Abstand des Elements zu anderen Elementen auf der Seite. Beschreiben Sie den Unterschied zwischen Padding und Margin.

## 1.5 Gesamtgroesse der Box

Messen oder schätzen Sie die effektive Breite der Box im Browser. Ist diese grösser als 300px? Begründen Sie Ihre Beobachtung.

## 1.6 box-sizing

Ergänzen Sie in Ihrer CSS-Datei folgende Regel ganz oben:

```
* {  
    box-sizing: border-box;  
}
```

Laden Sie die Seite neu. Was ändert sich an der Gesamtgroesse der Box? Erklären Sie den Unterschied zum vorherigen Verhalten.

## 1.7 Block-Elemente untersuchen

Erstellen Sie zwei weitere `div`-Elemente unterhalb der bestehenden Box, ebenfalls mit der Klasse `box`.

Beobachten Sie:

- Wie werden die Elemente untereinander angeordnet?
- Welche Breite nehmen die Elemente ein?

Warum ist dieses Verhalten typisch für Block-Elemente?

## 1.8 Inline-Elemente

Erstellen Sie innerhalb eines Paragrafen ein `span`-Element und geben Sie diesem die Klasse `inline-test`. Schreiben Sie etwas Text in das `span`.

Definieren Sie in der CSS-Datei folgende Eigenschaften für `.inline-test`:

- Hintergrundfarbe: gelb
- Breite: 200px
- Margin-top: 30px

Welche dieser Eigenschaften haben einen sichtbaren Effekt? Welche nicht? Begründen Sie Ihre Beobachtung.

## 1.9 Inline zu Block ändern

Ändern Sie nun den Display-Typ des `span`-Elements auf `block`.

`display: block;`

Was ändert sich am Verhalten des Elements? Welche CSS-Eigenschaften funktionieren nun wie erwartet?

## 1.10 inline-block

Setzen Sie den Display-Typ des `span`-Elements nun auf `inline-block`. Ergänzen Sie zusätzlich ein Padding von 10px.

Vergleichen Sie das Verhalten mit `inline` und `block`. In welchen Situationen könnte `inline-block` sinnvoll sein?

## 1.11 Navigation untersuchen

Untersuchen Sie die Links in Ihrer Navigation (`a`-Tags). Handelt es sich dabei standardmäßig um Inline- oder Block-Elemente?

Ändern Sie den Display-Typ der Links auf `block`. Beobachten Sie den Effekt auf:

- Klickfläche
- Anordnung

## **1.12 Layout-Debugging**

Öffnen Sie die zur Verfuegung gestellte HTML-Datei im Browser. Die Seite enthaelt mehrere absichtlich eingebaute Layout-Fehler. Ziel dieser Aufgabe ist es, diese Fehler zu erkennen, zu verstehen und zu beheben.

### **1.12.1 Navigation analysieren**

Betrachten Sie die Navigation im oberen Bereich der Seite.

- Wie verhalten sich die Links im Layout?
- Wie gross ist die Klickflaeche der einzelnen Links?

Warum ist dieses Verhalten typisch fuer den aktuellen Display-Typ der a-Tags?

### **1.12.2 Navigation verbessern**

Passen Sie das CSS so an, dass:

- jeder Link in einer eigenen Zeile dargestellt wird oder
- die Klickflaeche klar erkennbar und gross ist

Welcher Display-Typ eignet sich hierfuer besonders gut?

### **1.12.3 Box-Modell untersuchen**

Betrachten Sie die beiden Boxen im Hauptbereich der Seite.

- Welche Breite haben die Boxen effektiv im Browser?
- Entspricht diese Breite dem definierten Wert von 300px?

Begrunden Sie Ihre Beobachtung anhand des CSS Box-Modells.

### **1.12.4 Box-Modell korrigieren**

Passen Sie das CSS so an, dass die Gesamtbreite der Boxen tatsaechlich 300px betraegt, inklusive Padding und Border.

Welche CSS-Eigenschaft wird dafuer benoetigt?

### **1.12.5 Inline-Element analysieren**

Betrachten Sie das hervorgehobene Wort im Paragrafen.

- Welche CSS-Eigenschaften haben keinen sichtbaren Effekt?
- Warum ist das bei diesem Element so?

Beziehen Sie sich dabei auf den Display-Typ des Elements.

#### **1.12.6 Inline-Element korrigieren**

Andern Sie den Display-Typ des `span`-Elements so, dass:

- Breite und Höhe angewendet werden
- das Element weiterhin im Textfluss bleibt

Welcher Display-Wert eignet sich hierfür?

#### **1.12.7 Reflexion**

Beschreiben Sie in 2–3 Sätzen:

- einen Fehler, den Sie gefunden haben
- wie Sie diesen behoben haben
- welche Regel oder welches Konzept dahinter steckt

### **1.13 Bonus: Layout Experiment**

Experimentieren Sie mit folgenden CSS-Eigenschaften:

- Unterschiedliche Margin- und Padding-Werte
- Display-Typen bei verschiedenen Elementen
- Kombination von `width`, `height` und `display`

Notieren Sie sich mindestens zwei Beobachtungen, die Sie über das Layout-Verhalten von HTML-Elementen gemacht haben.