

Aufgabenblatt: Positionierung von Elementen mit CSS

Teil 1: Grundlagen verstehen

Beantworte die folgenden Fragen schriftlich:

- Was bedeutet `position: static;`?
- Welche CSS-Eigenschaften (`top`, `left`, `right`, `bottom`) haben bei `static` keine Wirkung?

Teil 2: Relative Positionierung

Gegeben ist folgendes CSS:

```
.box {  
  position: relative;  
  top: 20px;  
  left: 30px;  
}
```

- Beschreibe, von welchem Punkt aus das Element verschoben wird.
- Bleibt der ursprüngliche Platz des Elements im Dokument erhalten?

Erstelle ein eigenes Beispiel mit:

- einem Container
- einem relativ positionierten Element

Teil 3: Absolute Positionierung

Erkläre den Unterschied zwischen `position: relative;` und `position: absolute;`.

Gegeben ist folgender HTML-Code:

```
<div class="container">  
  <div class="box"></div>  
</div>
```

Und folgendes CSS:

```
.container {  
  position: relative;
```

```
}
```

```
.box {  
  position: absolute;  
  top: 10px;  
  right: 10px;  
}
```

- Relativ zu welchem Element wird die Box positioniert?
- Was passiert, wenn `position: relative;` beim Container entfernt wird?

Teil 4: Fixe Positionierung

Beschreibe in eigenen Worten, wie sich `position: fixed;` verhält.

- Was passiert beim Scrollen der Seite?
- Nenne zwei typische Einsatzgebiete für fixe Positionierung.

Erstelle ein Beispiel für:

- eine fixe Navigationsleiste
- ein fixes Informationsfeld

Teil 5: Z-Index verstehen

Gegeben sind zwei überlappende Boxen:

```
.box1 {  
  position: absolute;  
  z-index: 1;  
}
```

```
.box2 {  
  position: absolute;  
  z-index: 2;  
}
```

- Welche Box liegt im Vordergrund?
- Warum funktioniert `z-index` nur bei positionierten Elementen?

Erstelle ein eigenes Beispiel mit mindestens drei Elementen und unterschiedlichen `z-index`-Werten.

Teil 6: Fehler finden und korrigieren

Der folgende CSS-Code enthält konzeptionelle Fehler:

```
.box {  
  z-index: 10;  
  top: 20px;  
}
```

- Erkläre, warum der Code nicht wie erwartet funktioniert.
- Korrigiere den Code so, dass er sinnvoll eingesetzt werden kann.

Teil 7: Praxisaufgabe

Erstelle eine HTML-Seite mit folgenden Anforderungen:

- Ein Container mit relativer Positionierung
- Mindestens zwei absolut positionierte Elemente darin
- Ein fixes Element, das beim Scrollen sichtbar bleibt
- Überlappende Elemente mit sinnvoll eingesetztem `z-index`

Achte auf saubere Struktur und lesbaren Code.

Teil 8: Reflexion

Beantworte die folgenden Fragen:

- Welche Positionierungsart war am schwierigsten zu verstehen?
- In welchen Situationen würdest du absolute Positionierung vermeiden?
- Warum ist ein gutes Verständnis von Positionierung wichtig für Layouts?