

## ZUSAMMENFASSUNG CSS Grundkurs – Elemente mit CSS positionieren

### EINFÜHRUNG POSITIONIERUNG: DIE CSS EIGENSCHAFT POSITION

- Für die Positionierung von HTML-Elementen existiert die CSS Eigenschaft position
- Falls man die Eigenschaft position nicht explizit setzt, ist die Eigenschaft bei jedem HTML-Element von Beginn an auf den Wert static gesetzt
- static bedeutet, dass jedes Element, das auf static gesetzt ist, im normalen Elementfluss bleibt
- der normale Element-Fluss ist der, wie man Elemente im HTML-Dokument anordnet

### DIE WERTE RELATIVE UND ABSOLUTE

#### Position-Value relative

Relative verhält sich zunächst genau wie static, sodass divs normal im Elementfluss erscheinen. Wenn man den Wert auf relative gesetzt hat, kann man allerdings zusätzlich die vier weiteren CSS-Eigenschaften top, bottom, left und right setzen.

Mithilfe dieser Eigenschaften kann man das Element, welches mit relative positioniert wurde, in jede beliebige Richtung verschieben. Die Position des Elements kann also verändert werden relativ zur Position, an der sich das Element normalerweise im Elementfluss befinden würde.

Im folgenden Beispiel wird das div um 20 Pixel von links nach rechts verschoben.

Beispiel:

```
.child {  
    background-color: springgreen;  
    height: 50px;  
    width: 50px;  
    position: relative;  
    left: 20px;  
}
```

#### Position-Value absolute

Absolute sorgt dafür, dass das entsprechende Element komplett aus dem Elementfluss herausgehoben wird und alle anderen Elemente gerendert werden, als wäre das absolute position gesetzte Element gar nicht vorhanden. Auch hier können die CSS-Eigenschaften top, bottom, left und right verwendet werden.

Im folgenden Beispiel wird das div ganz oben am Rand des Bodys dargestellt.

Beispiel:

```
.child {  
    background-color: springgreen;  
    height: 50px;  
    width: 50px;  
    position: absolute;
```

```
    top: 0px;
}
```

## Unterschied zwischen relative und absolute

Der Unterschied zwischen relative und absolute ist der, dass relative positionierte Elemente relativ zu der Position verschoben werden, in welcher sie sich eigentlich im Element-Fluss befinden. Bei absolute positionierten Elementen ist dies nicht der Fall, denn diese verschieben sich „absolute“ zum nächstäußeren Container, der nicht auf den Position-Value static gesetzt ist.

## DER WERT FIXED

Der Wert fixed richtet sich automatisch am Viewport aus. Der **Viewport** ist die Bezeichnung für den Anzeigebereich, der gerade im Browserfenster sichtbar ist. Also der komplette **Sichtbereich** des **Browserfensters**. Im folgenden Beispiel richtet sich das div 0 Pixel von oben entfernt am Viewport aus.

Beispiel:

```
.child {
    background-color: springgreen;
    height: 50px;
    width: 50px;
    position: fixed;
    top: 0px;
}
```

## DER WERT STICKY

Das Element, welches auf sticky gesetzt wird, behält zunächst ganz normal seine Position im Element-Fluss. Wenn man allerdings scrollt und das obere Seitenende erreicht, bleibt das Element sozusagen „kleben“ und scrollt mit. Die Position, ab der ein Element kleben bleibt und mitscrollt, kann man über die Eigenschaften top, bottom, left und right definieren. Im folgenden Beispiel scrollt die Menüleiste immer dann mit, wenn ganz oben der Viewport erreicht ist.

Beispiel:

```
.main-nav {
    height: 40px;
    background-color: steelblue;
    text-align: center;
    padding: 20px;
    position: sticky;
    top: 0px;
}
```

## DER Z-INDEX

- Mit dem z-Index gibt man den Wert in z-Richtung an und bestimmt somit die Tiefe von Elementen
- Die Tiefe entscheidet, welches Element angezeigt wird, wenn zwei Elemente übereinander liegen
- Hat ein Element einen höheren z-Index als ein anderes und beide überlappen sich an einer Position, dann wird das Element mit dem höheren z-Wert oben angezeigt
- Wenn man den z-Index nicht explizit setzt, hat jedes Element per default einen z-Index von 0
- wenn sich Elemente an gewissen Stellen überlappen und alle Elemente den gleichen z-index aufweisen, ergibt sich auf Basis der sogenannten „stacking order“, welches Element über welchem anderen Element angezeigt wird
- die stacking order funktioniert so, dass das Element, das als letztes im Element-Fluss kommt, über allen anderen Elementen angezeigt wird
- das Element das hingegen als erstes im Element-Fluss kommt, wird ganz unten angezeigt

### WICHTIG

Der z-Index funktioniert nur, wenn der Position-Value **nicht** auf **static** oder **none** gesetzt ist.

Im folgenden Beispiel wird der z-Index auf den Wert 1 gesetzt.

Beispiel:

```
#div-2 {  
    background-color: green;  
    left: 75px;  
    bottom: 75px;  
    z-index: 1;  
}
```

### Stacking Context

- Der Stacking Context ist eine Gruppe von Elementen, die einen gemeinsamen parent haben und sich gemeinsam auf und ab auf der z-Achse verschieben