

# LU13d - CSS Transforms

Transforms sind geometrische Operationen, die das Aussehen eines Elements verändern, **ohne den Dokumentfluss zu beeinflussen** - andere Elemente rücken nicht nach. Sie werden sehr häufig innerhalb von @keyframes und Transitions eingesetzt.

## Die vier Grundoperationen

```
/* Verschieben: x = horizontal (positiv = rechts), y =  
vertikal (positiv = unten) */  
transform: translate(20px, 10px);  
transform: translateX(20px);    /* nur horizontal */  
transform: translateY(-10px);   /* nur vertikal, nach oben  
*/  
  
/* Skalieren: 1 = Originalgrösse, > 1 vergrössert, < 1  
verkleinert */  
transform: scale(1.5);          /* gleichmässig */  
transform: scale(1.2, 0.8);     /* horizontal x vertikal */  
  
/* Rotieren: positive Werte = im Uhrzeigersinn, Einheit deg  
immer nötig */  
transform: rotate(45deg);  
transform: rotate(-15deg);     /* gegen den Uhrzeigersinn  
*/  
  
/* Scheren: erzeugt einen Parallelogramm-Effekt */  
transform: skewX(15deg);        /* untere Kante verschoben  
*/  
transform: skewY(10deg);        /* rechte Kante verschoben  
*/
```

## Mehrere Transforms kombinieren

Mehrere Transformationen lassen sich in einer transform-Angabe kombinieren. **Die Reihenfolge beeinflusst das Ergebnis**, weil jede Transformation das Koordinatensystem für die nächste verändert:

```
/* Erst rotieren, dann verschieben:  
Die Verschiebung erfolgt im bereits rotierten  
Koordinatensystem */  
transform: rotate(45deg) translate(50px, 0);
```

```
/* Erst verschieben, dann rotieren:  
Die Rotation erfolgt um den Ursprung, nicht um das  
verschobene Element */  
transform: translate(50px, 0) rotate(45deg);
```

## transform-origin

Standardmässig liegt der Ursprungspunkt aller Transformationen in der **Mitte des Elements**. Mit transform-origin lässt er sich verschieben:

```
/* Rotation um die obere linke Ecke */  
transform-origin: left top;  
  
/* Rotation um die Mitte der Oberkante */  
transform-origin: center 0px;  
  
/* Rotation um einen Punkt ausserhalb des Elements */  
transform-origin: 50% 150%;
```

Der Ursprungspunkt bleibt an seiner definierten Position – er bewegt sich **nicht mit**, wenn das Element sich bewegt.

## Transforms in Keyframe-Animationen

Transforms sind der häufigste Baustein in @keyframes, weil der Browser sie direkt auf der Grafikkarte berechnen kann – ohne das Layout der gesamten Seite neu zu berechnen:

```
/* Beispiel: schwebendes Element */  
@keyframes bewegen-rotieren {  
  0%, 100% {  
    transform: translateY(0) rotate(-3deg);  
  }  
  50% {  
    transform: translateY(-12px) rotate(5deg);  
  }  
}  
  
.deko-element {  
  animation: bewegen-rotieren 4s ease-in-out infinite;
```

}

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

[https://wiki.bzz.ch/de/modul/m291/learningunits/lu13/theorie/d\\_transforms](https://wiki.bzz.ch/de/modul/m291/learningunits/lu13/theorie/d_transforms)Last update: **2026/05/31 23:25**