

## CSS-Transformationen

Mit der CSS-Eigenschaft `transform` lassen sich HTML-Elemente drehen, skalieren, verschieben und verzerren. Das kann nützlich sein, um z.B. kleine Korrekturen an der Position vorzunehmen. Wenn diese Transformationen mit CSS-Transitions (siehe weiter unten) gekoppelt werden, kann dies sehr effektiv sein.

```
.angewinkelt { transform: rotate(-20deg); }  
.eingeschoben { transform: translateX(1rem); }  
.abgehoben { transform: translateY(-3px); }  
.aufgeblasen { transform: scale(1.25); }
```

Der Wert rechts des Doppelpunkts beschreibt die Art der Transformation und das Ausmass (in Klammern).

Du kannst mehr als einen Wert angeben und auf diese Weise z.B. eine Rotation, eine Verschiebung und eine Skalierung kombinieren:

```
.ausgeflippt { transform: rotate(27deg) translate(-25px, 100px)  
scale(1.5); }
```

### Transform vs. Document Flow

Ändert sich durch eine Transformation die Position oder Dimension eines Elements, hat dies keinen Einfluss auf die umliegenden Elemente. Dafür kann es zu Überlappungen kommen.

Werden hingegen Eigenschaften wie `margin` oder `font-size` animiert, verschieben sich alle nachfolgenden Elemente (*Reflow*). Der Browser muss neue Positionswerte für die betroffenen Elemente errechnen, was bei 60 Frames pro Sekunde schnell sehr aufwendig wird. Es kommt zu Ruckeln, Erhitzung und Surren des Ventilators.

### Ausgangspunkt der Transformation

Für bestimmte Transformationen ist es wichtig, wo der Ausgangspunkt dafür liegt, z.B. bei Rotationen. Dieser lässt sich mit der Eigenschaft `transform-origin` einstellen.

```
.kreisel {  
  rotate(12turn);  
  transform-origin: center;  
}
```

## Opacity

Keine Transformation, aber eine sinnvolle Ergänzung: Mit der CSS-Eigenschaft `opacity` stellst du die Deckkraft eines HTML-Elements ein, von 0 (transparent) bis 1 (undurchsichtig).

```
.quasi-invisible { opacity: 0.1; }
```

Diese Eigenschaft passt gut zu den Transformationen, weil sie mit ihnen die Gruppe der **grundlegenden Animationen** bildet – wenn sie über eine gewisse Dauer interpoliert werden (*Interpolation* bedeutet die Veränderung eines Wertes entlang einer Skala).

## Animation von transform/opacity vs. Performance

Werden die Eigenschaften `transform` und/oder `opacity` animiert, kann der Browser dafür direkt auf die Hardware des Computers zurückgreifen: Die notwendigen Berechnungen werden von der Grafikkarte gemacht, die Software und der Arbeitsspeicher werden nicht tangiert. Es ruckelt nicht, das Gerät erhitzt sich nicht und der Ventilator beginnt nicht zu surren.