

1 Koordinatensystem

X-Achse links (-) / rechts (+)
Y-Achse vorne (-) / hinten (+)
Z-Achse unten (-) / oben (+)
Der Ursprung liegt bei [0, 0, 0].
Die Einheiten entsprechen üblicherweise Millimetern.

2 Vektoren lesen

Ein Vektor enthält Werte in der Reihenfolge X, Y, Z.
[20, 0, 5] bedeutet: 20 nach rechts und 5 nach oben.

```
position = [20, 0, 5];  
translate(position) cube([10,10,10]);
```

3 Zentriert oder ab Ursprung

center=false: Körper beginnt am Ursprung.
center=true: Körper liegt symmetrisch um den Ursprung.

```
cube([40, 20, 6], center = true);
```

4 Verschieben: translate()

Verschiebt einen Körper, ohne seine Form zu ändern.
Syntax: translate([x, y, z]) objekt;
Mehrere Achsen können gleichzeitig verwendet werden.

```
translate([25, 0, 8])  
  cube([20, 12, 6], center = true);  
  
// 25 mm nach rechts, 8 mm nach oben
```

5 Drehen: rotate()

Dreht um den Ursprung oder um den lokalen Ursprung.
Die Winkel werden in Grad angegeben.

[90,0,0] 90° um X
[0,90,0] 90° um Y
[0,0,45] 45° um Z

```
rotate([0, 0, 45])  
  cube([30, 10, 4], center = true);  
  
// Erst drehen, dann verschieben:  
translate([30, 0, 0])  
  rotate([0, 0, 45]) cube([30,10,4]);
```

6 Skalieren: scale()

Vergrößert oder verkleinert relativ zum Ursprung.
1 behält das Maß, 2 verdoppelt, 0.5 halbiert.
Ungleiche Faktoren verzerren den Körper.

```
scale([2, 1, 0.5])  
  sphere(d = 20);  
  
// Ergebnis: 40 × 20 × 10 mm
```

7 Spiegeln: mirror()

Spiegelt an einer Ebene durch den Ursprung.
Der Vektor beschreibt die Normale der Spiegelebene.

```
mirror([1, 0, 0]) // an YZ-Ebene  
  translate([20, 0, 0]) cube(10);
```

Merke

Eine Transformation wirkt auf den folgenden Körper oder Block.

8 Reihenfolge ist entscheidend

Verschachtelte Befehle wirken von innen nach außen.
Im ersten Beispiel wird erst gedreht, dann verschoben.
Im zweiten Beispiel wird erst verschoben, dann gedreht.

```
// A: Drehung am lokalen Ursprung
translate([30, 0, 0])
  rotate([0, 0, 45]) cube([20,8,4]);

// B: Bewegung um den Weltursprung
rotate([0, 0, 45])
  translate([30, 0, 0]) cube([20,8,4]);
```

9 Um einen eigenen Punkt drehen

1. Drehpunkt zum Ursprung verschieben.
 2. Körper drehen.
 3. Drehpunkt zurückverschieben.
- Das ist z. B. für Türen oder Hebel nützlich.

```
translate([drehpunkt_x, 0, 0])
  rotate([0, 0, winkel])
    translate([-drehpunkt_x, 0, 0])
      cube([40, 8, 4]);
```

10 Mehrere Körper gemeinsam bewegen

Geschweifte Klammern bilden einen Block.
Die davorstehende Transformation wirkt auf alle Inhalte.

```
translate([30, 0, 0]) {
  cube([20, 20, 4], center=true);
  translate([0, 0, 8]) sphere(d=10);
}
```

11 Symmetrische Bauteile

Eine Hälfte wird konstruiert und anschließend gespiegelt.
union() verbindet Original und Spiegelbild.
Die X-Position bleibt über eine Variable einstellbar.

```
abstand = 25;

module bohrung() {
  cylinder(h=8, d=5, center=true);
}

union() {
  translate([abstand,0,0]) bohrung();
  mirror([1,0,0])
    translate([abstand,0,0]) bohrung();
}
```

12 Typische Fehler

- X, Y und Z im Vektor vertauscht
- Drehung um die falsche Achse
- center=true nicht berücksichtigt
- Transformation durch Semikolon abgetrennt
- falsche Reihenfolge bei Verschachtelung

```
// Falsch:
translate([10,0,0]); cube(10);
// Richtig:
translate([10,0,0]) cube(10);
```

13 Debugging-Hilfen

- # hebt den transformierten Körper farbig hervor.
- % zeigt einen Referenzkörper transparent an.
- ! zeigt nur den markierten Körper.

```
%cube([60,40,2], center=true);
#translate([20,0,5]) sphere(d=10);
```

Mini-Übung

Erzeuge vier Zylinder bei $[\pm 25, \pm 15, 0]$.
Drehe danach das gesamte Lochbild um 30° um Z.