

1 Aufbau eines Befehls

Befehle enden normalerweise mit einem Semikolon.
Parameter stehen in runden Klammern.
Benannte Parameter machen Code leichter lesbar.

```
cube([40, 25, 6], center = true);  
cylinder(h = 12, d = 8, center = false);
```

2 Kommentare

Kommentare erklären Code, erzeugen aber keine Form.
// gilt bis zum Zeilenende.
/* ... */ eignet sich für längere Hinweise.

```
// Das ist ein kurzer Kommentar  
/* Mehrzeiliger  
   Kommentar */
```

3 Variablen

Werte erhalten einen eindeutigen Namen.
Änderst du den Wert, passt sich das Modell an.

```
breite = 60;  
hoehe = 35;  
staerke = 4;  
cube([breite, hoehe, staerke]);
```

4 Wichtige 3D-Grundkörper

Quader	<code>cube([x, y, z]);</code>
Kugel	<code>sphere(d = durchmesser);</code>
Zylinder	<code>cylinder(h = hoehe, d = durchmesser);</code>
Kegelstumpf	<code>cylinder(h=20, d1=18, d2=8);</code>
Zentrieren	<code>center = true</code>

```
cube([30, 20, 5], center = true);  
sphere(r = 10);  
cylinder(h = 15, d = 10);
```

5 Wichtige 2D-Grundformen

Rechteck	<code>square([breite, hoehe]);</code>
Kreis	<code>circle(d = durchmesser);</code>
Kontur	<code>polygon(points = [...]);</code>

2D-Formen besitzen noch keine Höhe.
Mit `linear_extrude()` werden sie zu 3D-Körpern.

```
linear_extrude(height = 5)  
square([40, 25], center = true);
```

6 Klammern und Blöcke

Runde Klammern () enthalten Parameter.
Eckige Klammern [] bilden Vektoren oder Listen.
Geschweifte Klammern { } fassen Befehle zusammen.

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

7 Groß-/Kleinschreibung

OpenSCAD unterscheidet Groß- und Kleinschreibung.
breite und Breite sind zwei verschiedene Namen.
Umlaute in Variablenamen besser vermeiden.

```
breite = 40; // empfohlen  
platten_hoehe = 5;
```

Merke

Ein fehlendes Semikolon ist ein häufiger Syntaxfehler.
Fehlermeldung und Zeilennummer stehen in der Konsole.

8 Koordinatensystem

X-Achse links / rechts
Y-Achse vorne / hinten
Z-Achse unten / oben
Der Ursprung liegt bei [0, 0, 0].
Positive Z-Werte zeigen nach oben.

9 Position und Drehung

`translate()` verschiebt einen Körper.
`rotate()` dreht in Grad um X, Y und Z.

```
translate([20, 0, 5])
  cube([10, 10, 10]);
rotate([0, 0, 45])
  cube([20, 8, 4]);
```

10 Ansicht bedienen

Linke Maustaste: Ansicht drehen
Rechte Maustaste: Ansicht verschieben
Mausrad: zoomen

11 Vorschau, Rendern, Export

F5 schnelle Vorschau
F6 vollständig rendern
Strg+S SCAD-Datei speichern
STL nach F6 exportieren
Vor dem Export immer vollständig rendern.
Die Konsole zeigt Warnungen und Fehler.

12 Qualität runder Körper

`$fn` legt die Anzahl der Kreissegmente fest.
Kleine Werte rechnen schnell, wirken aber kantig.
Für Vorschau und Rendern sind getrennte Werte möglich.

```
$fn = 64;
// Oder automatisch:
$fn = $preview ? 24 : 96;
cylinder(h = 10, d = 20);
```

13 Komplettes Beispiel: Grundplatte

Ziel: Eine zentrierte Platte mit variablen Abmessungen.
1. Werte festlegen
2. Körper mit Variablen erzeugen
3. F5 drücken und Werte ausprobieren

```
// Parametrische Grundplatte
breite = 80;
hoehe = 50;
staerke = 4;

$fn = $preview ? 24 : 96;

cube(
  [breite, hoehe, staerke],
  center = true
);
```

Mini-Übung

Ändere `breite` auf 100 und `staerke` auf 6.
Ergänze daneben einen Zylinder mit `d = 12` und `h = 20`.
Speichere das Modell als `grundplatte.scad`.