

Teil 2 -Beta – Aber trotzdem sehr gut!

WICHTIG: Der Download enthält „BOSL2-master“ mit dem gleichen Unterverzeichnis noch einmal! Nur diesen Inhalt nach BOSL2 kopieren.

Die Komplette Anleitung ist unter

Openscad -> libraries -> documentation -> Index by function-modul name:

<https://github.com/BelfrySCAD/BOSL2/wiki/AlphaIndex>

zu finden. Diese Seite ist alphabetisch angelegt und seeeeeeehr lang. Es lohnt sich trotzdem zu scrollen ...

Für mich das Beste an **BOSL2** ist die Möglichkeit Quadern, Zylindern und anderen Formen eine Fase oder einen Radius zu verpassen. Alles ohne großen Aufwand – meist nur in einer einzigen Zeile!

CUBOID

```
include <BOSL2/std.scad>
```

```
// Einfacher Quader
```

```
cube([30,40,50], center=true);
```

```
// Quader mit Fase von 5
```

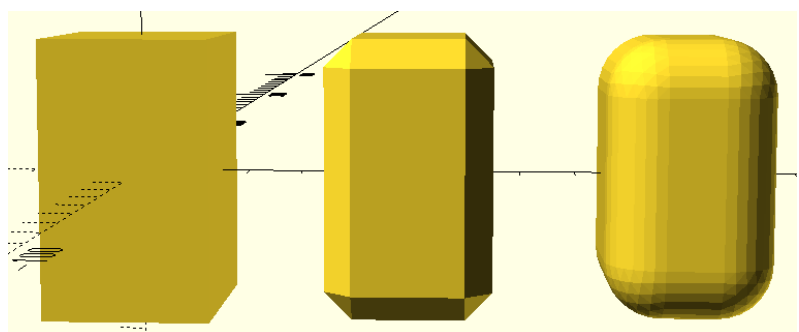
```
translate([50,0,0])
```

```
cuboid([30,40,50], chamfer=5);
```

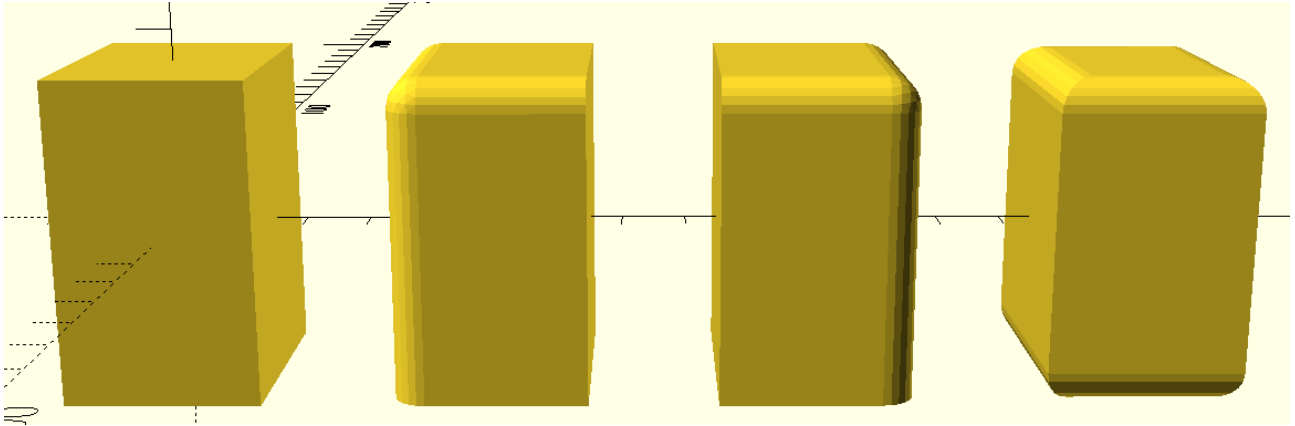
```
// Quader mit Radius 10
```

```
translate([100,0,0])
```

```
cuboid([30,40,50], rounding=10);
```



Anstatt „cube“ wird nun „cuboid“ angegeben. „chamfer“ entspricht der Fase und „rounding“ dem Radius.



```
include <BOSL2/std.scad>

// Einfacher Quader
cube([30,40,50], center=true);

// Quader mit Fase von 5
translate([50,0,0])
cuboid([30,40,50], rounding=5,
  edges=[TOP+FRONT, TOP+LEFT, FRONT+LEFT],
  $fn=24);

// Quader mit Radius 10
translate([100,0,0])
cuboid([30,40,50], rounding=5,
  edges=[TOP+FRONT, TOP+RIGHT, FRONT+RIGHT],
  $fn=24);

// Quader mit Radius 10- Oben und unten Radius
translate([150,0,0])
cuboid([30,40,50], rounding=5,
  edges=[BOTTOM, TOP],
  $fn=24);
```

In dieser Variante können die Radien (rounding) oder Fasen (chamfer) nur an bestimmten Stellen erscheinen.

TOP = die Oberseite / BOTTOM = Unterseite / FRONT = Vorn / BACK = Hinten

FRONT+LEFT, FRONT+RIGHT, FRONT+TOP, FRONT+BOTTOM sind die Angaben der Vorderseite.

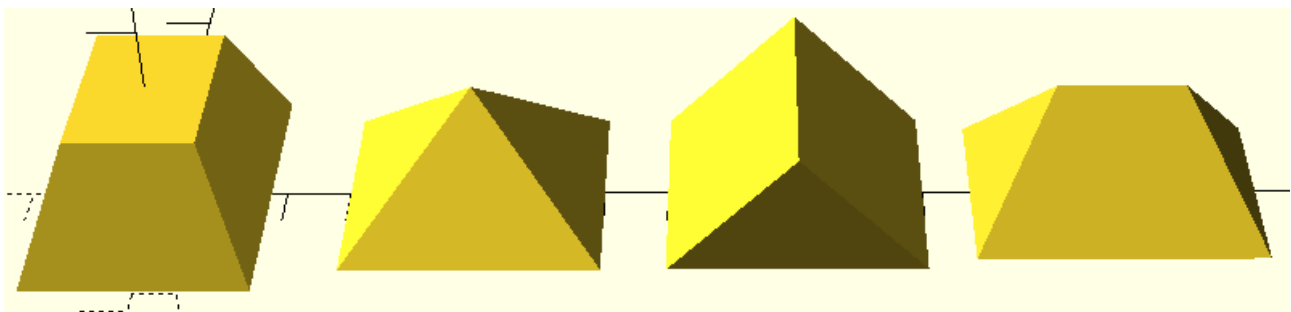
BACK+LEFT, BACK+RIGHT, BACK+TOP, BACK+BOTTOM die der Rückseite.

TOP+LEFT, TOP+RIGHT, TOP+FRONT, TOP+BACK die der Oberseite.

BOTTOM+LEFT, BOTTOM+RIGHT, BOTTOM+BACK, BOTTOM+FRONT die der Unterseite.

Zu beachten ist die Großschreibung und das PLUS dazwischen.

Das PRISMOID



1. Es werden die Ausmaße des unteren Bodens, des oberen Rechteckes und die Höhe angegeben.
2. Bei den oberen Werten = 0, fallen sie zusammen und es entsteht eine Pyramide.
3. Wird nur ein Wert =0, entsteht ein Hausdach.
4. Bei unterschiedlichen Größenangaben von unten und oben, gleichzeitig ein oberer Wert = 0, kommt ein Hausdach mit Schrägen heraus.

// Funktion: prismoid

include <BOSL2/std.scad>

prismoid(size1=[35,50], size2=[20,30], h=20);

// Quader mit Fase von 5

translate([50,0,0]) prismoid([40,40], [0,0], h=20);

// Quader mit Radius 10

translate([100,0,0]) prismoid(size1=[40,40], size2=[0,40], h=20);

// Quader mit Radius 10

translate([150,0,0]) prismoid(size1=[45,35], size2=[20,0], h=20);

22-04

size1 = entspricht den unteren Maßangaben und size2 den oberen.

Verschieben der Mittellinie über SHIFT

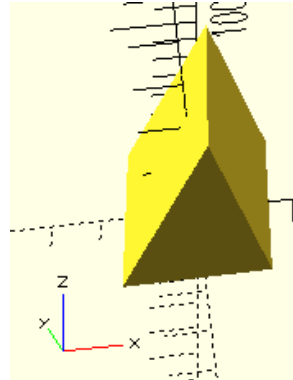
```
include <BOSL2/std.scad>
```

```
prismoid(size1=[30,60],
```

```
size2=[0,60],
```

```
shift=[5,0],
```

```
h=30);
```



Wird der prismoid-Befehl mit shift ergänzt, kann die obere Fläche/Mittellinie verschoben werden. Mit positiven Werten wandert es in den Plus-Bereich von Z.

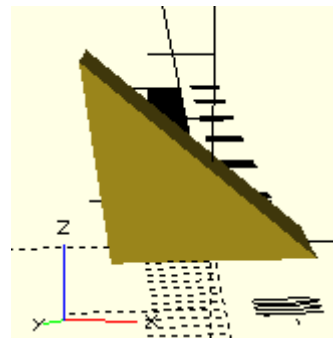
```
include <BOSL2/std.scad>
```

```
prismoid(size1=[30,60],
```

```
size2=[0,60],
```

```
shift=[-20,0],
```

```
h=30);
```



Bei negativen Werten kann ein Überhang erzeugt werden.

Mit dem zweiten Wert von shift wird es in der Y-Achse verschoben:

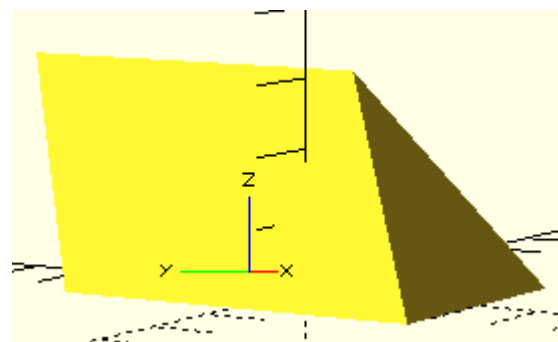
```
include <BOSL2/std.scad>
```

```
prismoid(size1=[30,60],
```

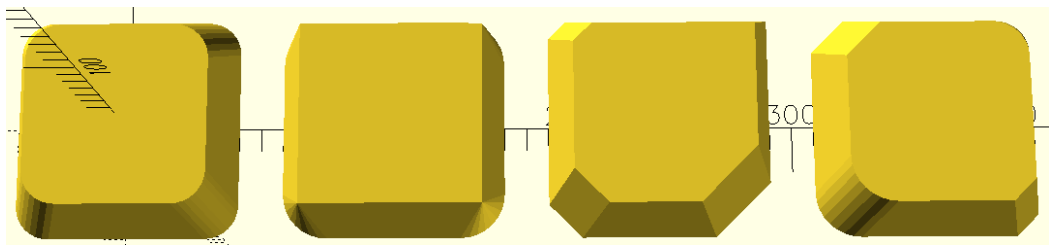
```
size2=[0,60],
```

```
shift=[10,30],
```

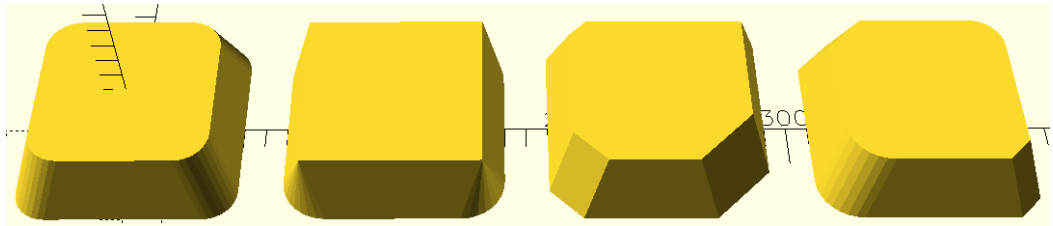
```
h=30);
```



Die vier
Elemente
(einmal
Draufsicht
und einmal
von vorn)



zeigen die
Möglichkeiten
der Fasen und
Radien an.



```
include <BOSL2/std.scad>
```

```
prismoid(100, 80, rounding=15, h=30);
```

```
translate([120,0,0]) prismoid(100, 80, rounding1=20, rounding2=0, h=30);
```

```
translate([240,0,0])prismoid(100, 80, chamfer=[0,8,15,25], h=30);
```

```
translate([360,0,0])prismoid(100, 80, h=30,  
chamfer=[0,15,0,10],  
rounding=[15,0,20,0]);
```

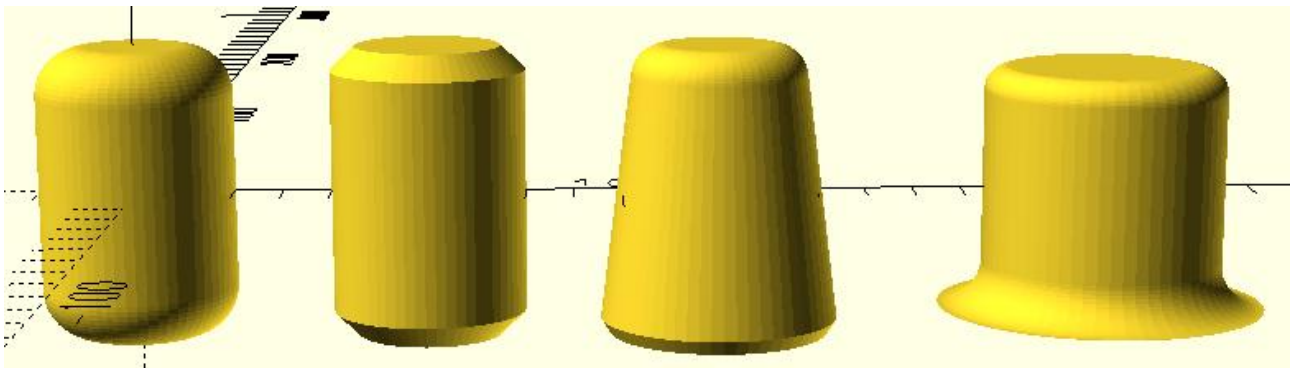
Das erste Objekt erhält durch „rounding=15“ an allen vier Ecken einen Radius mit 15.

Das zweite Teil wurde durch „rounding1“ und „rounding2“ ergänzt. Hier können unterschiedliche Radien angegeben werden, sogar bis Null.

Das dritte Element erhielt „chamfer=[0,8,15,25]“ zu den Angaben der einzelnen Kanten. Es wird von oben rechts gegen den Uhrzeigersinn gezählt. Oben rechts = 0. Oben links erhält 8, unten links 15 und unten rechts 25.

Nummer 4 ist die totale Kombination. Chamfer=[0,15,0,10] bedeutet, dass oben links 15 und unten rechts 10 die Fase wird. Rounding=[15,0,20,0] dagegen teilt die obere rechte Kante mit 15 ein und die untere Linke mit 20.

Zylinder mit Fase oder Radius versehen



```
include <BOSL2/std.scad>
$fn=60;

cyl(l=60, d=40, rounding=10);

translate([60,0,0]) cyl(l=60, d=40, chamfer=5);

translate([120,0,0])
cyl(
  l=60, d1=50, d2=35,
  chamfer1=3, chamfang1=55,
  from_end=true, rounding2=7
);

translate([200,0,0]) cyl(l=50, r=25, rounding1=-9, rounding2=5);
```

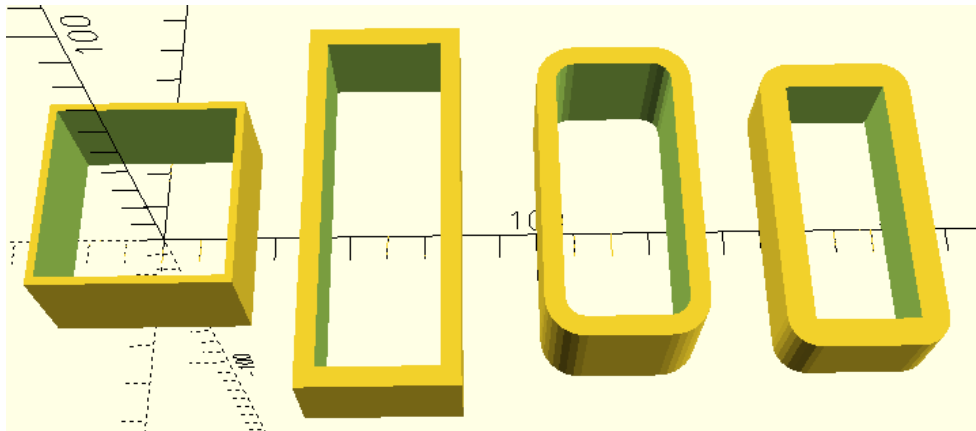
Der erste Zylinder erhält über „rounding“ den Radius, der zweite über „chamfer“ seine Fase.

Der dritte ist ein Kegel, allein durch die zwei Maßangaben von d1 und d2. „chamfer1“ bezeichnet die untere Fase, wobei „chamfang“ den Innenkreis bestimmt. Mit „rounding2“ wird der obere Radius festgelegt.

Der vierte Knopf ist das Absolute: Über „rounding1“ mit **negativem** Wert erstellt die Rundung nach außen. „rounding2“ dagegen lässt oben den Radius erscheinen.

Fehlt irgendwo ein Drehknopf?

RECHTECK-ROHRE



```
include <BOSL2/std.scad>
```

```
rect_tube(size=50, wall=2, h=30);
```

```
translate([60,0,0]) rect_tube(size=[40,100], wall=5, h=25);
```

```
translate([120,0,0]) rect_tube(size=[40,80], wall=5, rounding=10, h=30);
```

```
translate([180,0,0]) rect_tube(size=[40,80], wall=7, rounding=8, irounding=0, h=20);
```

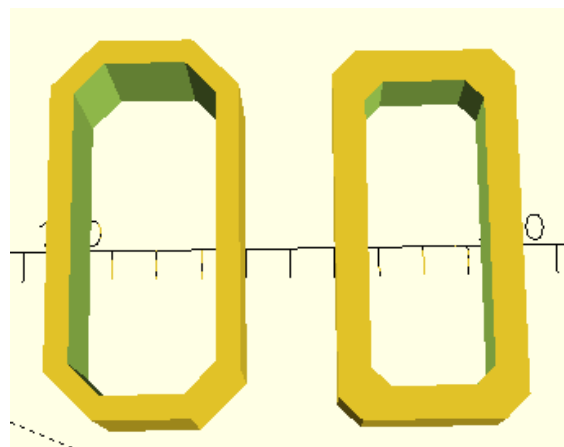
1. Das quadratische Rohrstück erhält das Außenmaß (size) und die Wandstärke (wall). Die Höhe dürfte bekannt sein.
2. „size=[40,100]“ bestimmt das Außenmaß.
3. „rounding=10“ beschreibt den äußeren Radius.
4. Mit „irounding=0“ wird der innere Radius festgelegt. „0“ ergibt eine scharfe Innenkante. Der Wert kann ebenso positive Werte annehmen, um einen Innenradius zu erhalten.

Gleichermaßen mit „chamfer“ (Fase) möglich:

```
include <BOSL2/std.scad>
```

```
rect_tube(size=[40,80],  
wall=5, chamfer=10, h=30);
```

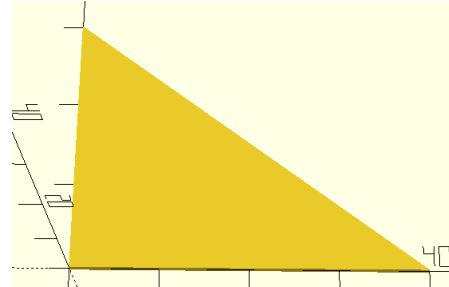
```
translate([60,0,0])  
rect_tube(size=[40,80],  
wall=7, chamfer=5,  
ichamfer=4, h=20);
```



SCHNELLES DREIECK

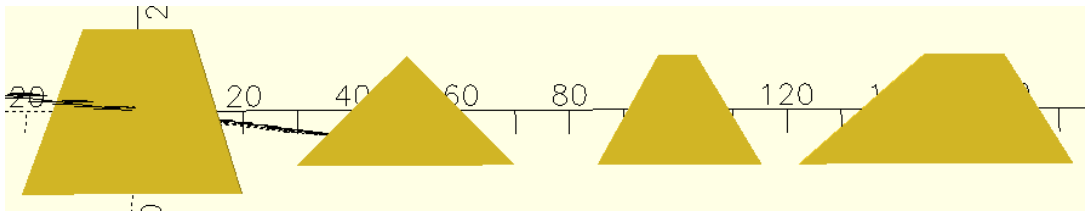
Standard in 2D-Formen zum fertigen Dreieck in einer Zeile:

```
include <BOSL2/std.scad>
right_triangle([40,30]);
```



Nein, ein „left_triangle“ fand ich nicht.

TRAPEZOID



```
include <BOSL2/std.scad>
```

```
trapezoid(h=30, w1=40, w2=20);
```

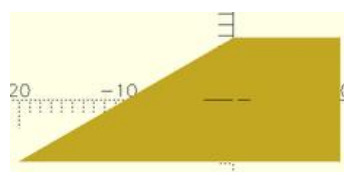
```
translate([50,0,0]) trapezoid(h=20, w1=40, w2=0);
```

```
translate([100,0,0]) trapezoid(h=20, w1=30, ang=60);
```

```
translate([150,0,0]) trapezoid(h=20, w1=50, ang=[40,60]);
```

1. Mit „w1“ und „w2“ werden die Breiten des Trapezoids erstellt. „h“ ist die Höhe.
2. Hier ist der obere Wert gleich Null = ein Dreieck entsteht.
3. Es wird die untere Breite und Höhe angegeben. Mit „ang“ der Winkel festgelegt.
4. „ang=[40,60]“ bestimmen nun den linken und den rechten Winkel.

Bei 90° wäre dies eine senkrechte Wand.
„ang=[30,90]“



„chamfer“
kann auch
hier
eingesetzt
werden:



```
include <BOSL2/std.scad>
```

```
trapezoid(h=30, w1=60, w2=40, chamfer=5);
```

```
translate([70,0,0]) trapezoid(h=30, w1=60, w2=40, chamfer=-5);
```

```
translate([140,0,0]) trapezoid(h=30, w1=60, w2=40, chamfer=-5, flip=true);
```

Sowie

„rounding“



```
include <BOSL2/std.scad>
```

```
trapezoid(h=30, w1=60, w2=40, rounding=5);
```

```
translate([70,0,0]) trapezoid(h=30, w1=60, w2=40, rounding=-5);
```

```
translate([140,0,0]) trapezoid(h=30, w1=60, w2=40, rounding=-5, flip=true);
```

Bei **positiven** Werten von „chamfer“ und „rounding“ werden die Kanten entsprechend beschnitten.

Bei **negativen** Werten gehen die Abschnitte nach links und rechts Außen.

Mit „**flip=true**“ nach oben und unten.

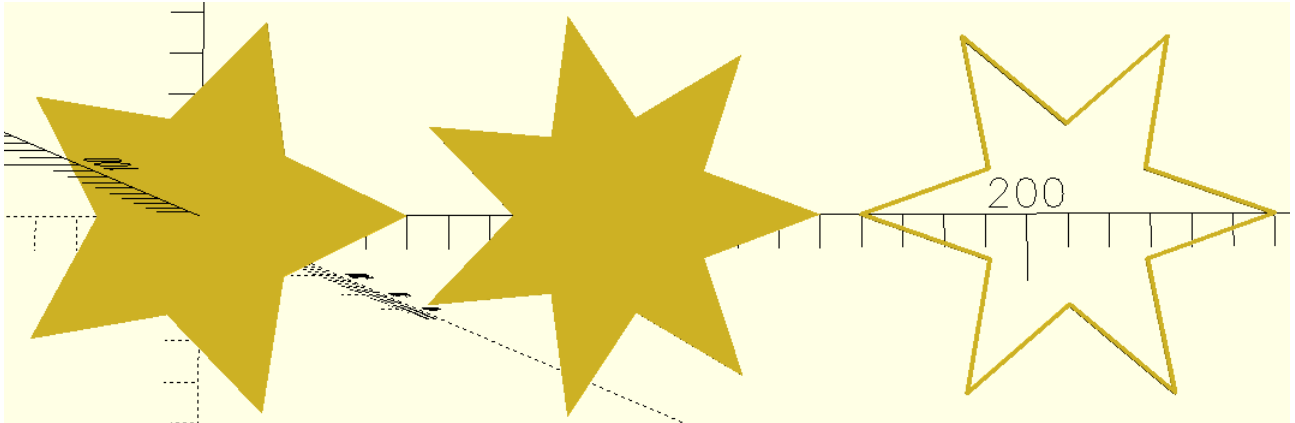
Kombinationen sind ebenso möglich:

```
include <BOSL2/std.scad>
trapezoid(h=30, w1=60, w2=40,
rounding=[5,0,-10,0],
chamfer=[0,8,0,-15],
$fa=1,$fs=1);
```



22-10

In der Weihnachtszeit fragt jeder nach Sternen ... Drucken Sie doch einfach welche!



```
include <BOSL2/std.scad>
```

```
star(n=5, r=50, ir=25);
```

```
translate([100,0,0]) star(n=7, r=50, step=2.7);
```

```
translate([210,0,0]) stroke(closed=true, star(n=6, r=50, ir=22));
```

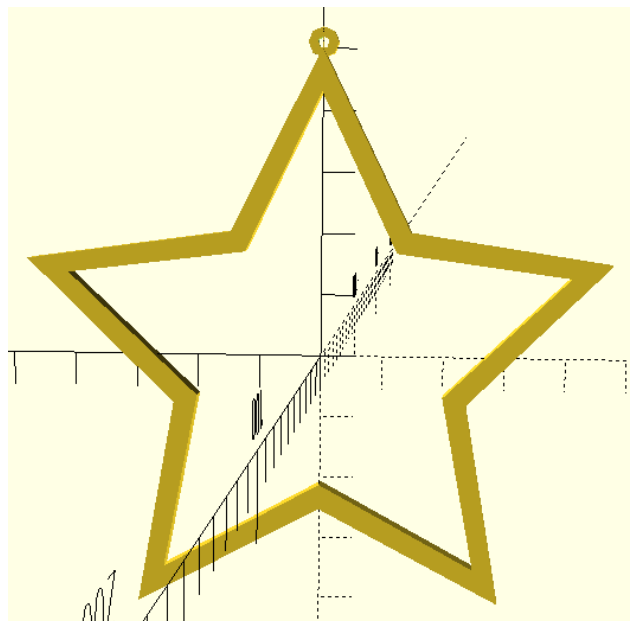
1. Unter „n“ wird die Anzahl der Strahlen angelegt. „r“ ist der äußere Durchmesser, „ir“ der Innere.
2. Hier kommt „step“ zum Einsatz! Bei minimaler Wertänderung erscheint der Stern neu. Geht man zu weit, ist er weg! Dran denken: Punkt statt Komma.
3. Ideal zum Lasern. Oder um etwas hineinzuhängen.

Der Weihnachtsstern mit Öse

zum Aufhängen ist unter

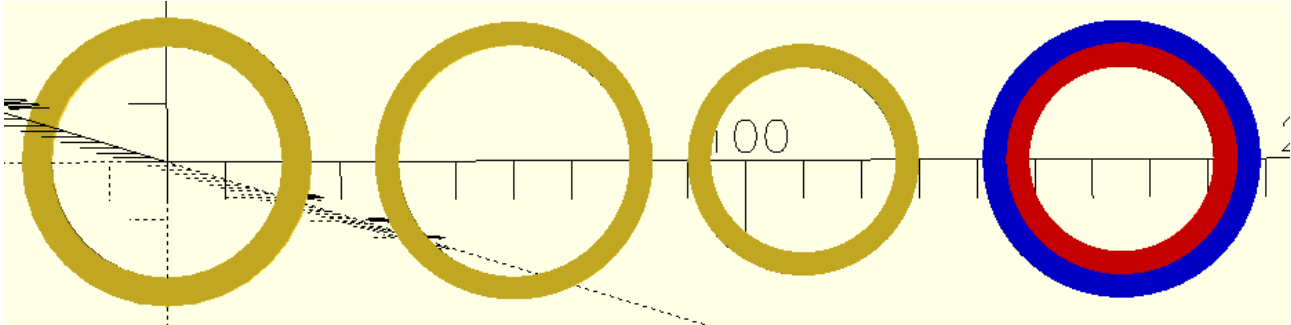
022-10-Weihnachtsstern.scad

zu finden.



RINGELREIHEN

Praktisch zu gestalten: Ein Ring oder eine Öse in nur einer einzigen Zeile.



```
include <BOSL2/std.scad>

ring(r1=20,r2=25, n=32);

translate([60,0,0]) ring(r=20,ring_width=4, n=50);

translate([110,0,0]) ring(r=20,ring_width=-4, n=50);

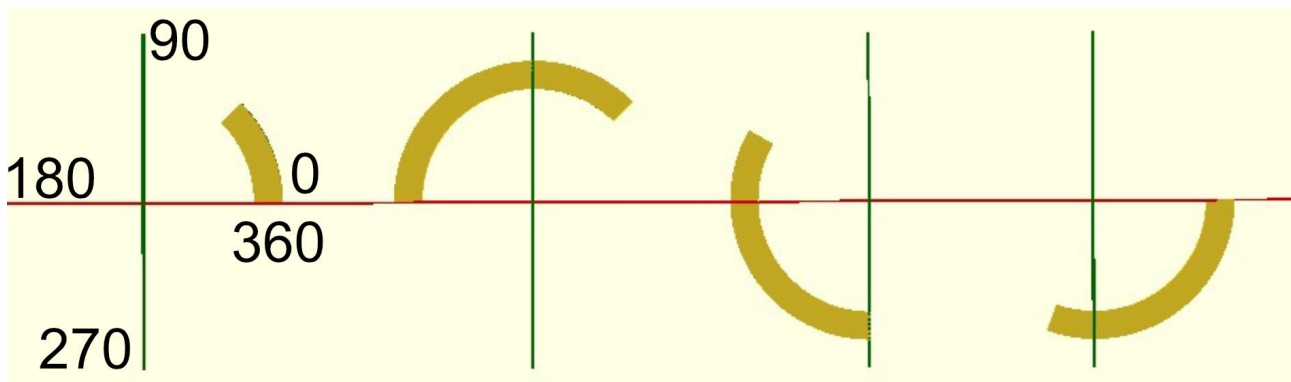
translate([165,0,0])color("blue",1) ring(r=20,ring_width=4, n=50);

translate([165,0,0])color("red",1) ring(r=20,ring_width=-4, n=50);
```

1. Mit dem Befehl „ring“ wird ein Ring erzeugt. „r1“ gibt den Innendurchmesser an und „r2“ den äußeren. „n“ ist wieder die Anzahl der Segmente zur Rundung.
2. „ring(r=20,ring_width=4“ gibt den Durchmesser an. Der positive Wert gibt immer die Ringstärke nach **außen** an.
3. „ring(r=20,ring_width=-4“ gibt den Durchmesser an. Der negative Wert gibt immer die Ringstärke nach **innen** an.
- 4 + 5. Hier wurden die beiden vorhergehende Ringe ineinander gefügt. Dem Blauen sein Innendurchmesser ist dem Roten sein Außendurchmesser.

Gleich ausprobieren mit dem Weihnachtsstern der Seite zuvor.

RINGABSCHNITTE



```
include <BOSL2/std.scad>
```

```
ring(r1=20,r2=25, angle=[0,45], n=50);
```

```
translate([70,0,0]) ring(r1=20,r2=25, angle=[45,180], n=50);
```

```
translate([130,0,0]) ring(r1=20,r2=25, angle=[150,270], n=50);
```

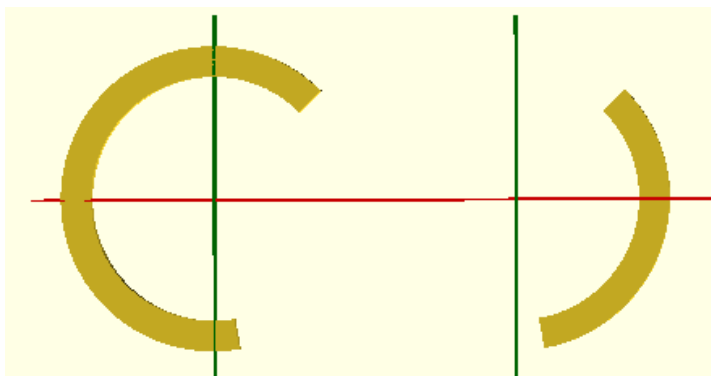
```
translate([170,0,0]) ring(r1=20,r2=25, angle=[250,360], n=50);
```

Wie sonst unter „ring“ werden auch hier „r1, r2“ und „n“ angegeben. Zusätzlich wird „angle=[0,45]“ eingefügt. Diese bezeichnen die Winkelangaben in Grad. Der erste Wert ist der Startpunkt, der zweite der Endpunkt.

Rechts ist immer 0°, oben 90°, links 180° und rechts wiederum 360°.

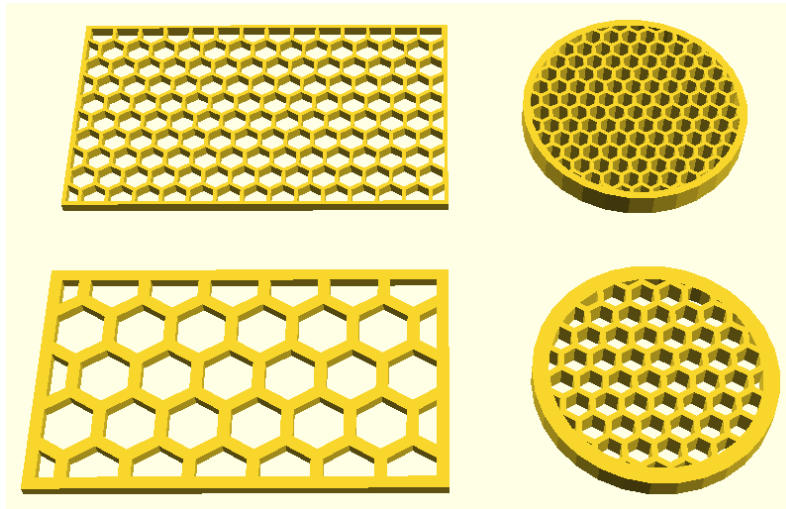
Wird ein Teilstück des Ringes von z.B. 280° nach 45° benötigt, so sind zwei Ringsegmente notwendig.

Ansonsten wird von 45° nach 280° das Segment erzeugt.



Erstes Stück von 280° nach 360°. Und das Zweite von 0° nach 45° ergibt das gewünschte Ergebnis.

LÜFTUNGSGITTER



```
include <BOSL2/std.scad>
include <BOSL2/walls.scad>
```

```
hex_panel([100, 60, 3], strut=1, spacing=7);
```

```
translate([0,-80,0]) hex_panel([100, 60, 3], strut=3, spacing=15);
```

```
translate([100,0,0]) hex_panel(circle(30), 1, 5, h = 7, frame = 2);
```

```
translate([100,-80,0]) hex_panel(circle(30), 1.5, 7, h = 6, frame = 4);
```

1. Rechteck (Links oben)

„hex_panel([100, 60, 3])“ gibt die Außenmaße des Rahmens an

„strut=1“ ergibt die Rahmenstärke sowie die der Waben

„spacing=7“ legt das Innenmaß der Waben fest.

2. Rechteck (links unten)

Rahmen- und Wabenstärke beläuft sich auf 3. Die Innenräume sind 15 groß.

3. Rund (oben rechts)

(circle(30), = Durchmesser des Teiles

1, = Wabenwandstärke

5, = Innenmaß der Waben

h = 7, = Höhe des Teiles

frame = 2); = Rahmenstärke außen

4. Rund (unten rechts) siehe bei 3.