

8.6 Aufgabe 10

Im Bild bei 8.3 wurde ein Quader mit einem Zylinder geschnitten und so die Oberfläche abgerundet. Mit einer Gleichung aus dem Bereich der [Formeln zum Kreissegment](#) kann man zu jedem beliebigen Quader die Werte des Zylinders bestimmen, die man zum Abrunden benötigt:

$$r = \frac{4h^2 + s^2}{8h}$$

Konstruiere einen allgemeinen Quader und schneide ihn mit dem dazu passenden Zylinder um ihn abzurunden:

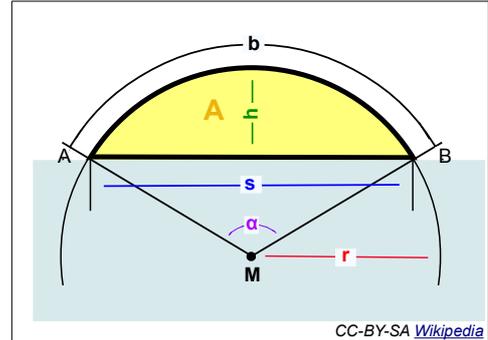
```
// r = (4 * h^2 + s^2) / (8 * h)

h=10; // Höhe des Quaders
s=50; // Breite des Quaders
l=100; // Länge des Quaders

// Radius des Schnittzylinders
radius = (4*h*h + s*s) / (8*h);

intersection()
{
    cube([l, s, h]);

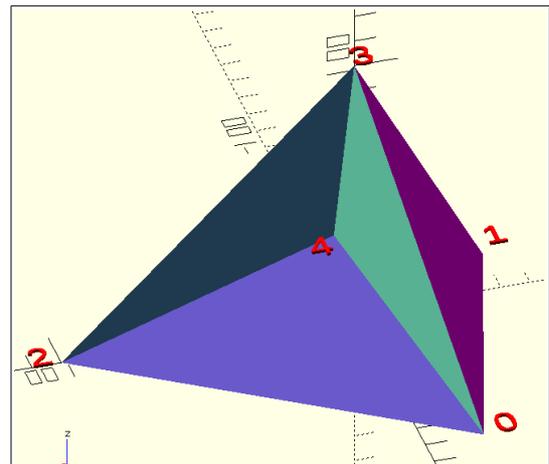
    translate([_____, _____, _____])
        rotate([0, 90, 0])
            cylinder(d=2 * radius, h=_____, $fn=100);
}
```



9. Polyeder

Beliebige Polyeder lassen sich mit der `polyhedron` Anweisung erzeugen. Als Argumente übergibt man Listen von Punkten und Flächen. Dabei benötigt jede Fläche drei Punkte und jeder Punkt besteht aus einer x-, y- und z-Koordinate. Die Zählung der Punkte beginnt bei 0.

```
polyhedron(points=[ [-20, 80, 0], // 0
                    [-50, -20, 0], // 1
                    [100, 0, 0], // 2
                    [ 0, 0, 100], // 3
                    [ 10, 10, 40] ], // 4
           faces=[ [0, 1, 2],
                  [0, 1, 3],
                  [1, 2, 3],
                  [0, 2, 4],
                  [0, 3, 4],
                  [2, 3, 4] ]]);
```

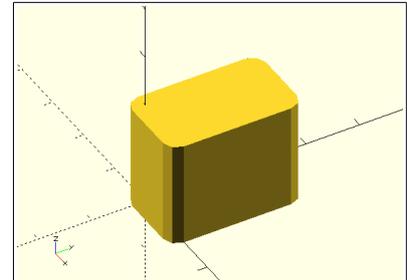


10. Weitere Transformationen

- In der Vorschau sind Farben möglich: `color("Farbe")` färbt das nachfolgende Objekt ein. Die möglichen Farbwerte sind [hier](#) aufgeführt.
- Objekte können mit `scale([x-Faktor, y-Faktor, z-Faktor])` vergrößert oder verkleinert werden.

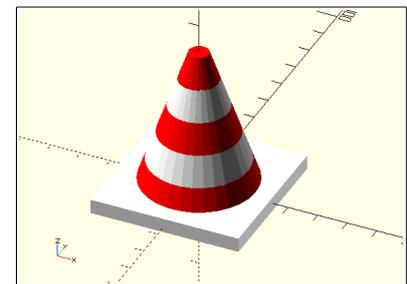
- Objekte können mit `resize([x-Größe, y-Größe, z-Größe])` auf die gewünschte Zielgröße gebracht werden.
- Eine Spiegelung an einer Ebene durch den Ursprung erreichst Du mit `mirror([x, y, z])`. Dabei gibt der Vektor die Normale zur Ebene an.
- Du kannst mit der `hull()` Anweisung mehrere Objekte in eine gemeinsamen Hülle stecken.
- Eine Minkowski-Summe erzeugt die Anweisung `minkowski()`. Dabei werden die Dimensionen der Objekte ebenfalls addiert. Etwas Eckiges und etwas Rundes wird dabei zu einem abgerundetem Eckigen:

```
minkowski()
{
    cube([10, 20, 10]);
    cylinder(d=5, h=10);
}
```



10.1 Aufgabe 11

Bearbeite die Lösung zur Aufgabe 2 und färbe den Verkehrs-Leitkegel rot/weiß ein.

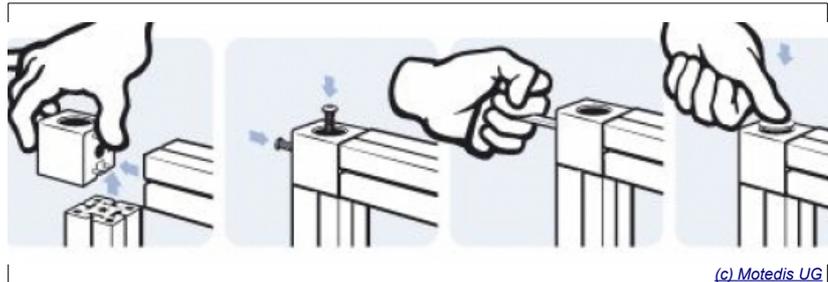


10.2 Aufgabe 12

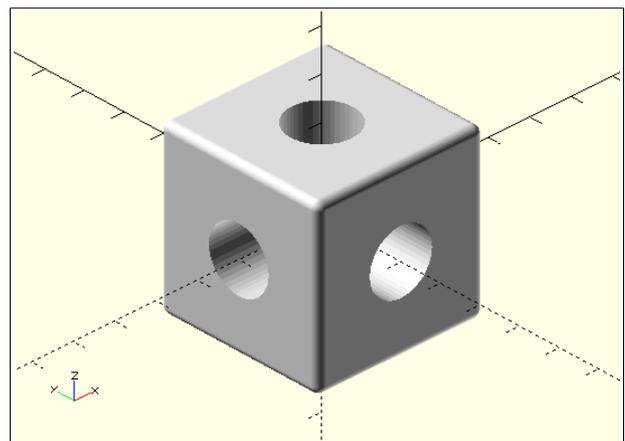
Das OpenLab benötigt ein 3D-Drucker Rack. Es soll neben Aluminium Strangprofilen aus Würfelverbindern aufgebaut sein.

Um eine Stückliste zu generieren, wird eine OpenSCAD Konstruktion des Racks benötigt. Konstruiere einen dunkelgrauen Würfelverbinder mit der Kantenlänge 40mm. Runde alle Kanten und Ecken mit einem Radius von 2mm ab (`minkowski`).

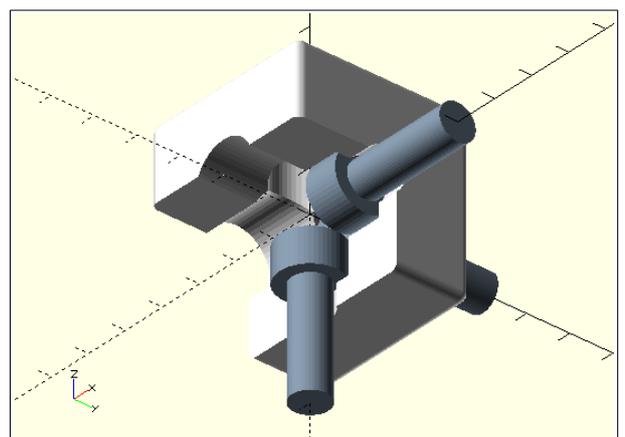
Zur Verbindung mit den Strangprofilen werden M8x25 Zylinderkopf Innensechskant Schrauben verwendet. Erzeuge passende Bohrungen, so dass man von drei Seiten die Schrauben einsetzen und mit den Profilen verschrauben kann. Die passenden Bohrdurchmesser (für den Schraubenkopf und das Gewinde) kannst Du aus dem Internet besorgen.



(c) Motedis UG



Bonusrunde: Die echten Würfelverbinder sind so konstruiert, dass sich die Schraubenköpfe im Inneren nicht gegenseitig behindern. Konstruiere eine Schraube mit annähernden Abmessungen und setze drei davon in deinen Würfelverbinder ein. Prüfe, ob sich die Schraubenköpfe berühren.



11. 2D Objekte

2-dimensionale Objekte werden von OpenSCAD in der Vorschau immer 3-dimensional angezeigt. Es funktionieren auch alle Transformationen – sie bleiben aber dennoch 2-dimensional. Du musst sie extrudieren um sie 3-dimensional zu bekommen.

- `circle(d=Durchmesser)`; erzeugt einen Kreis.
- `square(breite, länge)`; erzeugt ein Rechteck.
- `polygon(points=[Punkte], paths=[Pfade])`; erzeugt ein Vieleck aus den angegebenen Punkten und Pfaden. Die Nummer des ersten Punkts ist 0.

```

polygon(points=[ [0, 0], [10, 0], [10, 10] ],
        paths=[ [0, 1, 2] ]); // ein Dreieck

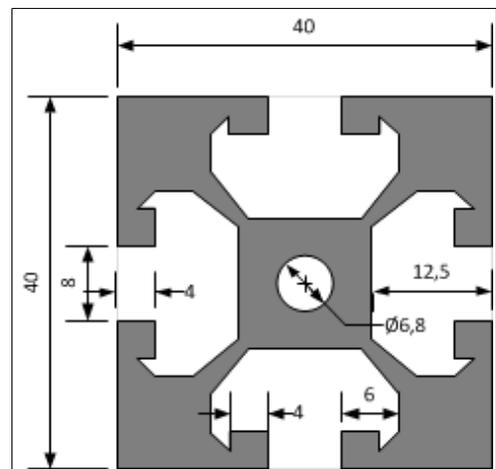
```

- `text("Beschriftung")`; erzeugt einen Text. Neben dem Textstring können noch Font, Größe, Ausrichtung etc. angegeben werden. Weitere Erläuterungen findest Du [hier](#).

11.1 Aufgabe 13

Das OpenLab benötigt ein 3D-Drucker Rack. Es soll hauptsächlich aus Aluminium Strangprofilen und Verbindern aufgebaut sein. Um eine Stückliste zu generieren, wird eine OpenSCAD Konstruktion des Racks benötigt. Konstruiere mit 2D Objekten den Querschnitt des Profils anhand dieser Zeichnung. Nicht eingezeichnete Maße sind Vielfache von 2 und dürfen geschätzt werden:

Tipp: Du brauchst nur etwa $\frac{1}{4}$ des Strangprofils zu konstruieren. Die anderen $\frac{3}{4}$ erzeugst Du durch Rotieren oder Spiegeln. Versuche beide Möglichkeiten!



12. Extrudieren von 2D-Objekten.

Um ein 2D-Objekt in die dritte Dimension zu bekommen, gibt es in OpenSCAD 2 Extrudier-Anweisungen:

- `linear_extrude(height=Höhe)` extrudiert das Objekt geradlinig. Es können auch Parameter für Drehung, Vergrößerung bzw. Verkleinerung angegeben werden. Weitere Details stehen [hier](#).
- `rotate_extrude()` extrudiert das Objekt während einer 360° Rotation.

12.1 Aufgabe 14

Das in Aufgabe 13 erzeugte 2D Objekt soll nun in die dritte Dimension gebracht werden. Erzeuge ein hellgraues Teststück des Alu Strangprofils in der Länge 100mm.

