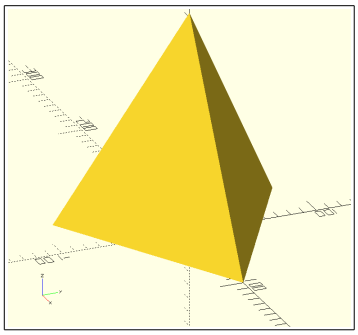
6. Prisma / Pyramide



In OpenSCAD kann man mit einem Trick Prismen erzeugen, deren Grundflächen regelmäßige Vielecke sind. Sie sind Spezialfälle von Zylindern. Mit dem Parameter **`$fn=x`** gibt man die Anzahl der Facetten eines runden Objekts an – und damit auch die Anzahl der Kanten. Ein 10-eckiges Prisma wird somit durch `cylinder(d=durchmesser, h=höhe, $fn=10);` erzeugt. Schiefe Prismen (auch mit unregelmäßigen Vielecken in der Grundfläche) folgen zu einem späteren Zeitpunkt.

Pyramiden sind Spezialfälle von Kegeln. Auch hier bestimmt der Parameter **`$fn=x`** die Anzahl der Facetten.

7. Konstanten und Berechnungen

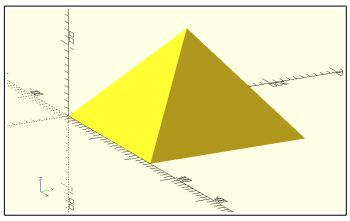
Sobald man in einer Konstruktion einen Messwert mehr als einmal verwendet, ist es schwierig, diesen Wert bei einer Änderung auch an allen verwendeten Stellen mit zu ändern. Deshalb ist es sinnvoll, möglichst alle Werte in Konstanten zu speichern. Verwende möglichst kleine Buchstaben und _ um die Namen zu bilden. Verzichte auf Umlaute und dergleichen:

```
hoehe=150;
breite_unten=10;
winkel=17.5;
beschriftung="Ich bin Text.";

rotate([winkel, 0, 0])
      cube([breite_unten, breite_unten, hoehe]);
```

OpenSCAD stellt eine Vielzahl von (mathematischen) Funktionen und Operationen zur Verfügung. Man kann entweder die Ergebnisse von Berechnungen in Konstanten speichern, oder die Berechnung an der Stelle einbauen, wo man sie benötigt. Werte in Konstanten können nicht verändert werden! Für Viele, die bereits programmieren können, ist dieses Konzept ein wenig verwirrend. Vielleicht wird es klarer, wenn man sich vorstellt, dass die Wertezuweisung zum Zeitpunkt des compilierens (also immer dann, wenn man F5 oder F6 drückt) stattfindet, nicht zur Laufzeit – wie bei normalen Programmen. Für die nächsten Aufgaben sind die Grundrechenarten sowie die Potenz- und Wurzelfunktion nötig:

```
// Satz des Pythargoras
// a² + b² = c²; c = √(a² + b²)

a=120;
b=230;
c=sqrt(pow(a, 2) + pow(b, 2));

rotate([winkel, 0, 0])
      cube([breite_unten, c - 25.7 + hoehe / 2, hoehe]);
```

7.1 Aufgabe 6



Erzeuge eine Cheops-Pyramide im Maßstab 1:1000 (1mm entspricht dann 1m in der Wirklichkeit). Die Maße der Pyramide findest Du in der Wikipedia. Beachte, dass die Seitenlänge der Pyramide nicht dem Durchmesser entspricht, den du bei `cylinder()` angeben musst. Wende den Satz des Pythargoras an, um den richtigen Durchmesser zu ermitteln. Setze die Pyramide so in das Koordinatensystem, dass zwei der Grundlinien auf der x- bzw. y-Achse verlaufen.

7.2 Aufgabe 7

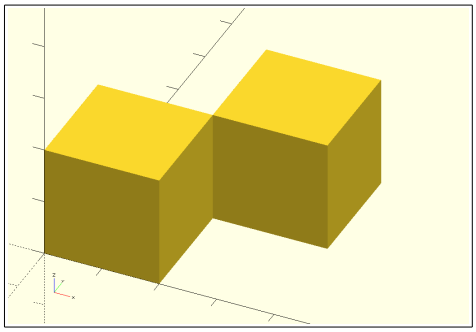
Suche die Formel zur Berechnung der Höhe h eines Tetraeders mit der Kantenlänge a. Baue einen Tetraeder mit einer Kantenlänge 100mm.

8. Operationen aus der Mengenlehre

8.1 Vereinigungsmenge

Einige der bisherigen Beispiele und Aufgaben verwendeten die von OpenSCAD immer automatisch durchgeführte Operation der Vereinigungsmenge `union()`. Wenn sich zwei einzelne Objekte überlappen, wird dadurch ein neues Gesamtobjekt gebildet. Du musst aber sicherstellen, dass es tatsächlich eine Überlappung gibt. Die folgenden Anweisungen machen dies deutlich:

```
laenge=20;
breite=laenge;
hoehe=laenge;
versatz_x=20;
versatz_y=20;
versatz_z=0;

// keine Überlappung
cube([breite, laenge, hoehe]);
translate([versatz_x, versatz_y, versatz_z])
      cube([breite, laenge, hoehe]);
```



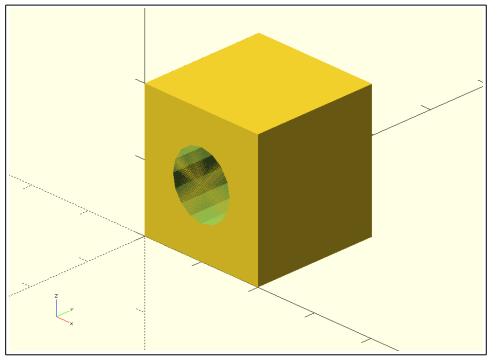
Der zweite Würfel teilt sich eine Kante mit dem ersten Würfel. Drücke jetzt F6 und betrachte die Fehlermeldung in der Konsole: `WARNING: Object may not be a valid 2-manifold and may need repair!` Hier kann der Renderer nicht entscheiden welchem Körper die Kante gehört. Es ist zwar möglich, eine .stl-Datei zu exportieren, diese muss aber vom Slicer repariert werden. Besser ist es, eine minimale Veränderung am Versatz einzubauen:

```
// minimale Überlappung
cube([breite, laenge, hoehe]);
translate([versatz_x - 0.01, versatz_y - 0.01, versatz_z])
      cube([breite, laenge, hoehe]);
```

8.2 Teilmenge

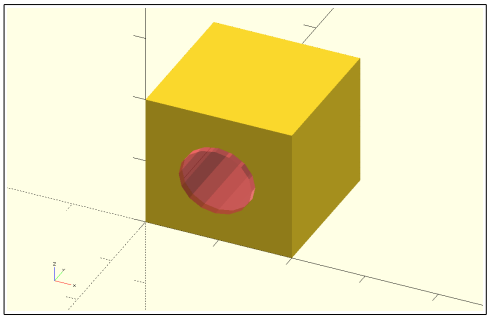
Eine ähnliche Problematik zeigt sich bei Teilmengen `difference()`. Man muss immer einen minimal größeren Teil herausschneiden als man eigentlich benötigt. Eine Bohrung in einem Würfel macht dies deutlich:

```
difference()
{
      cube([breite, laenge, hoehe]);

      translate([laenge/2, 0, hoehe/2])
            rotate([-90,0,0])
                  cylinder(d=10, h=breite);
}
```



Man sieht im Bild, dass die Oberfläche des Würfels nicht vollständig durchbohrt wurde. Bohre wie im richtigen Leben ganz durch den Körper durch. Der Debug-Modifier `#` zeigt dir die Bohrung in transparentem rot:

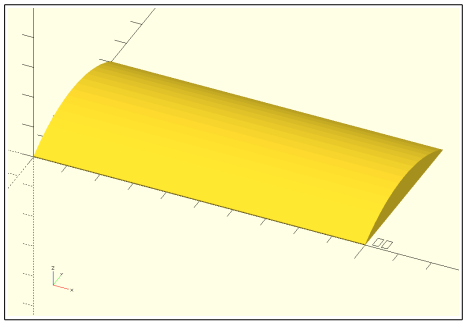
```
difference()
{
      cube([breite, laenge, hoehe]);

      translate([laenge/2, -1, hoehe/2])
            rotate([-90,0,0])
                  #cylinder(d=10, h=breite + 2);
}
```



Ausnahmen sind natürlich Sacklöcher und dergleichen, die absichtlich nicht ganz durch einen Körper gehen.

Hier sind im Objektinneren natürlich die exakten Maße einzuhalten.

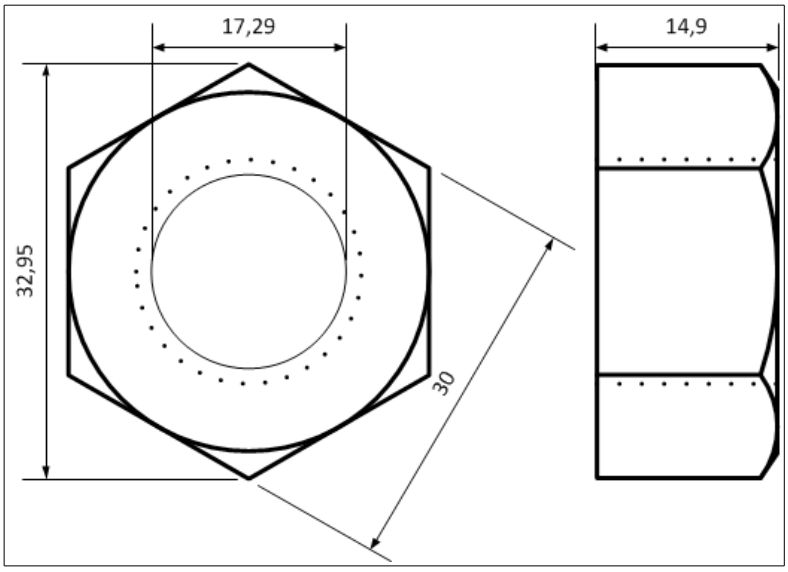
8.3 Schnittmenge



Die letzte verbleibende Mengenperation ist die Schnittmenge `intersection()`. Bei überlappenden Objekten bleibt nur der Teil übrig, der sich überlappt. Benutze auch hier den Debug-Modifier um den Quader bzw. den Zylinder sichtbar zu machen:

```
intersection()
{
cube([100, 50, 10]);

translate([0,25,-43])
      rotate([0,90,0])
            cylinder(d=100, h=110, $fn=100);
}
```

8.4 Aufgabe 8



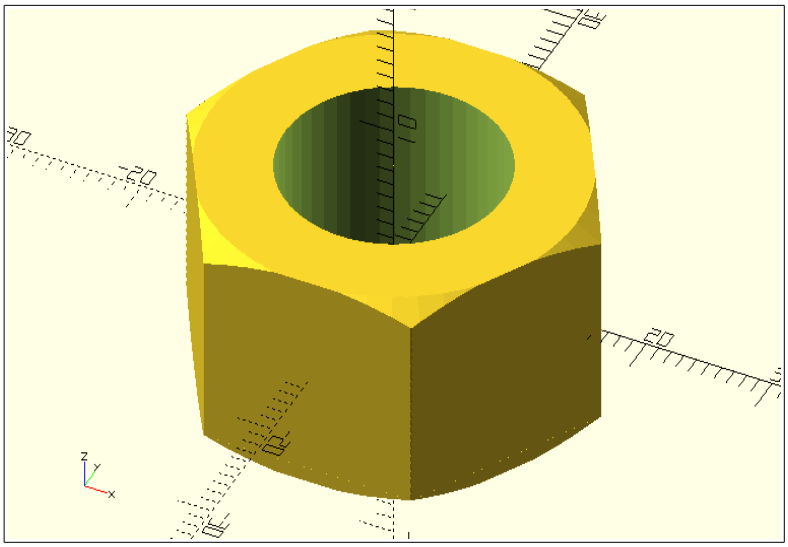
Es wird eine Sechskantmutter M20 benötigt. Die Werkstatt hat den Gewindebohrer schon in der Maschine. Liefere den Schlossern einen Rohling!

Auch hier kannst Du mit den Modifiern arbeiten um Teile hervorzuheben oder sichtbar machen.

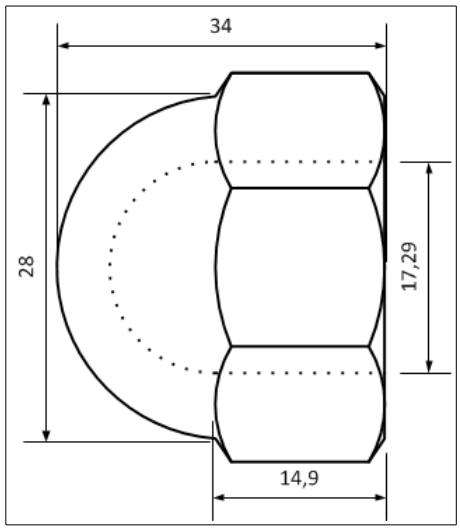
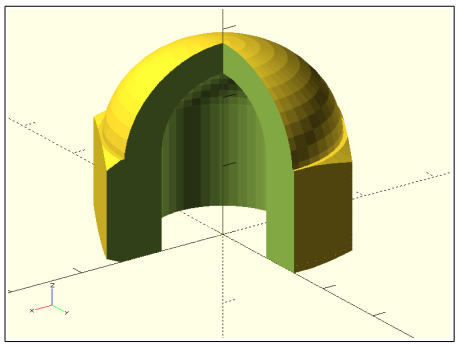
Tipp: Die typischen Fasen oben und unten an der Mutter erzeugst Du z.B. mit einer Schnitt- oder Teilmenge aus dem Sechskant-Prisma und einem Kegel. Die Fase soll einen Winkel von 30° zur Waagerechten haben. Die Höhe des Kegels ergibt sich über den Sinus des Fasenwinkels. Die Kegelkonstruktion sieht dann z.B. so aus:

```
fasenwinkel=30;
e=32.95;
h=(e/2) * sin(fasenwinkel);

cylinder(d1=e, d2=0, h=h);
```

Die untere Fase wird durch einen umgedrehten gleichen Kegel erzeugt.



8.5 Aufgabe 9

Schon wieder die Schlosser. Dieses mal wird zu Ausbildungszwecken eine Hutmutter M20 im Halbschnitt benötigt.

Beim Halbschnitt wird ¼ des Objekts weggeschnitten, so dass man das Innere sehen kann.