$fn= Anzahl der Fragmente ; **Kurven glätten/runden**

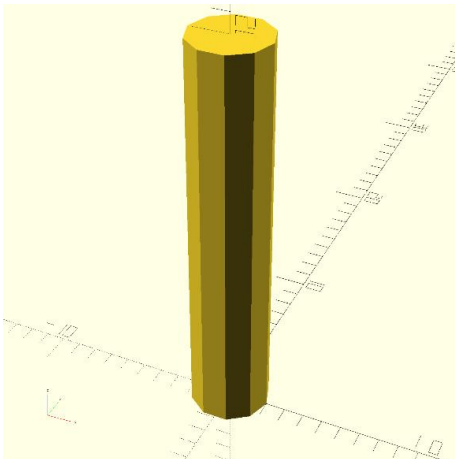
Warum wurden bisher Kugeln und Zylinder gezeichnet die nicht so ganz rund sind, sondern aus einer Reihe von flachen Paneelen bestehen?
Das liegt daran, dass OpenSCAD, wie die meisten 3D-Konstruktionsprogramme, eine Anzahl von geraden Linien verwendet, um sich einer Kurve anzunähern. Um Speicher zu sparen und die Verarbeitungszeit zu reduzieren, die zum Zeichnen komplexer Formen erforderlich ist, verwendet OpenSCAD normalerweise eine relativ kleine Anzahl dieser Zeilen.
Der abgebildete Zylinder im Bild verwendet zum Beispiel nur sechs Liniensegmente, um sich der Kurve der Kreisfläche des Zylinders anzunähern.

Selbstverständlich kann die Anzahl angegeben werden, um die Zylinder und Kugeln glatter zu gestalten!

Um eine Kurve runder zu gestalten, wird einfach der $fn-Parameter in der gewünschten Form eingeschlossen. Oder er wird an den Anfang des Scriptes gesetzt und gilt für alle Formen …
Wird beispielsweise $fn auf 10 gesetzt, sieht der Zylinder sofort etwas runder aus. (weil es den Umfang des Zylinders mit 10 Liniensegmenten zeichnet)

```
cylinder(h=20, r=2, $fn=10);
```

Bild:



Erhöhung der Liniensegmente auf 10 über den Befehl „$fn=10“, der in den Zeichenbefehl eingefügt wurde

Wie bei anderen Parametern wird $fn in Klammern innerhalb eines Befehles eingefügt.

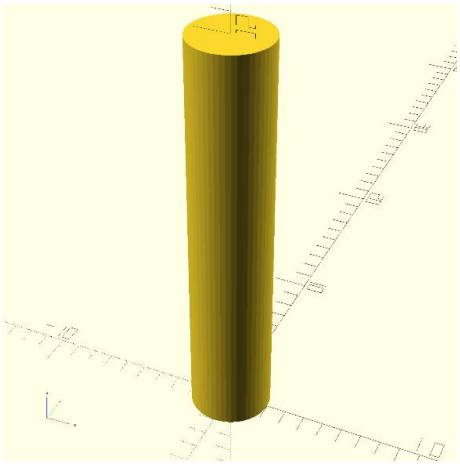
Obwohl der Zylinder im Bild runder wirkt als ein Standardzylinder, erscheint er immer noch nicht richtig rund. Wird $fn auf einen größeren Wert eingestellt, wird ebenso der Zylinder noch runder dargestellt.

```
cylinder(h=20, r=2, $fn=50);
```

004-02

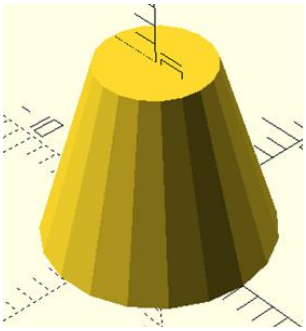
Mit 50 Liniensegmenten sieht die Außenkurve in diesem Zylinder viel glatter aus. Ab einem bestimmten Punkt zeigt die Erhöhung von $fn jedoch keine sichtbaren Auswirkungen mehr.

Achtung: OpenSCAD benötigt mit großen $fn-Werten mehr Zeit, um die Formen mit noch mehr Details zu generieren.

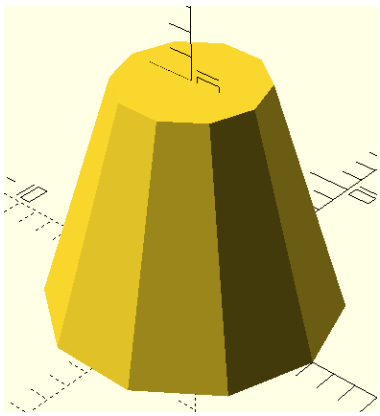


Wenn ein passender Wert für $fn gesucht wird, wird es immer ein Kompromiss zwischen Glätte und Rechenaufwand sein. Im Allgemeinen erzeugt **$fn=50** eine annehmbare größere „Rundheit“, die als ausreichend bewertet werden kann.

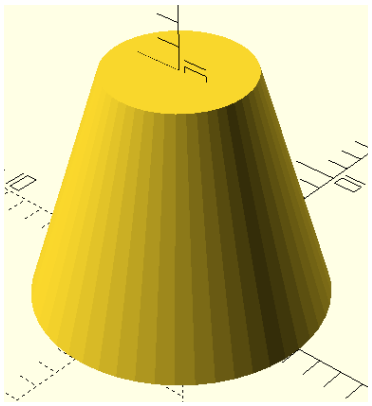
Bei einer Eingabe von cylinder(h=10, r1=6, r2=3);
also ohne $fn=… erscheint dieses Gebilde.
Dies entspricht einem Wert von 19



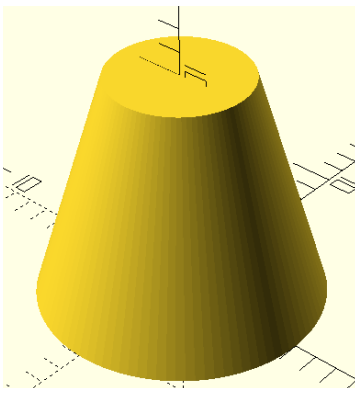
$fn=10;
cylinder(h=10, r1=6, r2=3);



$fn= 50;

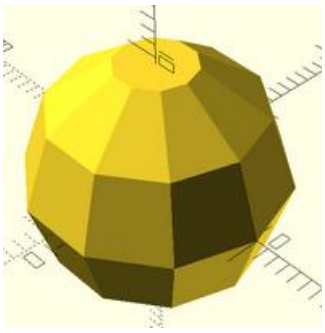
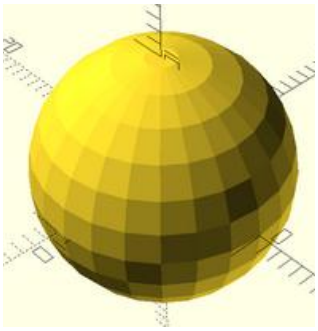
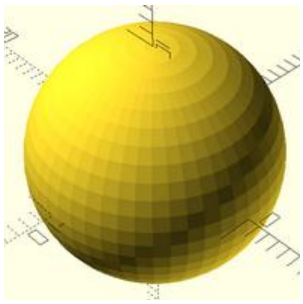


$fn= 100;

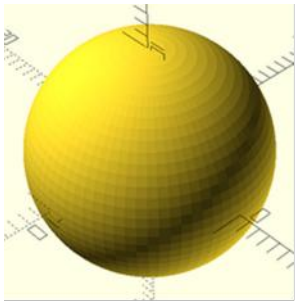
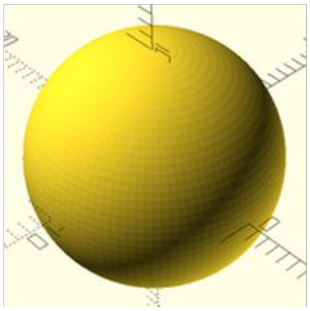
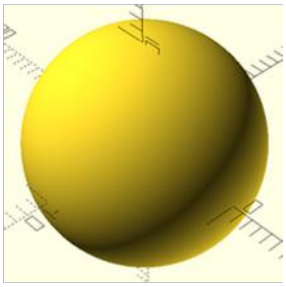


Am deutlichsten sichtbar wird dies bei einer Kugel:

$fn= 10;** **$fn= 25; **$fn= 50;**

$fn= 75;** **$fn= 100; **$fn= 300;**

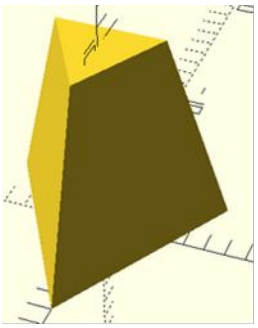
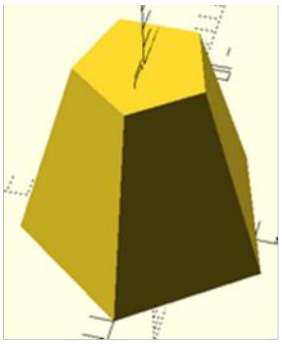
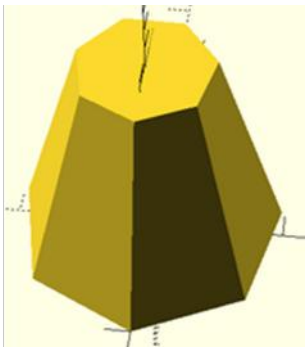




Zur Erstellung von Objekten genügen meistens Werte zwischen 40 und 60. Zum Rendern kann der Wert erhöht werden – je nach System und Rechenzeit ……

Dadurch können z.B. auch diverse Objekte erstellt werden

$fn=3;** **$fn=5; **$fn=7; usw…..**

cylinder(h=10,r1=6,r2=3);

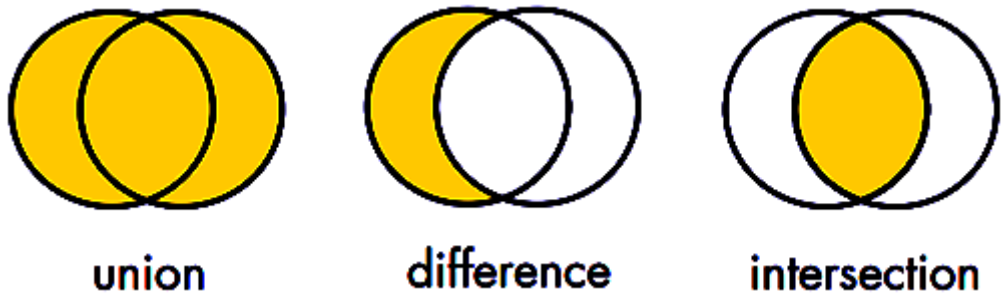




004-04

Kombinieren von 3D-Formen mit booleschen Operationen

Manchmal möchte man Formen mit Funktionen erstellen, die komplexer sind als die Grundformen, die bisher erstellt wurden. Mit den booleschen Operationen in OpenSCAD werden mehrere Formen, wie Quader, Kugeln, Zylinder und Kegel in eine einzige Form kombiniert.

Dies wird durch Verwendung einer von den drei Möglichkeiten erreicht: Vereinigung, Differenz oder Schnittmenge.



Der Befehl „union“ gruppiert zwei Formen zu einer gemeinsamen Form,
der Befehl „difference“ subtrahiert eine Form von einer anderen und
„intersection“ behält nur die Teile, an denen sich zwei Formen schneiden.

Boolesche Befehle

Boolean wird als ja/nein vorgestellt. Es wird häufig in Mathematik und Software verwendet, wenn definiert werden muss, ob etwas existiert oder nicht.
Die hier besprochenen booleschen Operationen definieren, ob ein Raumvolumen Material enthalten oder leer sein soll.

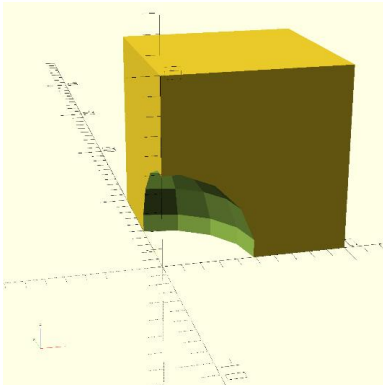
difference ( ) { ….. }

Formen von einander subtrahieren

Es wird mit der Subtraktion von Formen begonnen mit dem Befehl „difference“

```
difference() {
cube([10, 10, 10]);
sphere(5);
}
```

Von einem Würfel wird eine Kugel subtrahiert



Der Befehl lautet:

difference() {

……

}

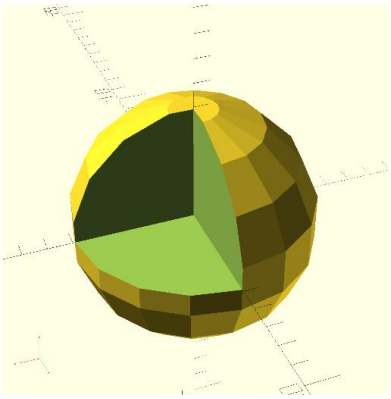
Es wird „difference mit zwei runden Klammern () angegeben, gefolgt von zwei geschweiften Klammern{ … } . Zwischen { und } kommen die Formen.
Die **erste Form ist das Grundelement**, von dem alle nachfolgende Formen subtrahiert werden.

Was passiert, wenn beide Formen aus Versehen vertauscht werden?

```
difference() {
sphere(5);
cube([10, 10, 10]);
}
```

Das erste Element ist die Kugel. Von ihr wird nun der Würfel subtrahiert.

Das Umkehren der Operationen erzeugt eine Kugel mit genau einer fehlenden Ecke, wobei „cube“ eine Würfelform auf die ursprüngliche Kugel gezeichnet hätte.



004-06

Geisterbild / Debuggen von „difference“

Debuggen = entlausen -> Computerfehler entfernen.

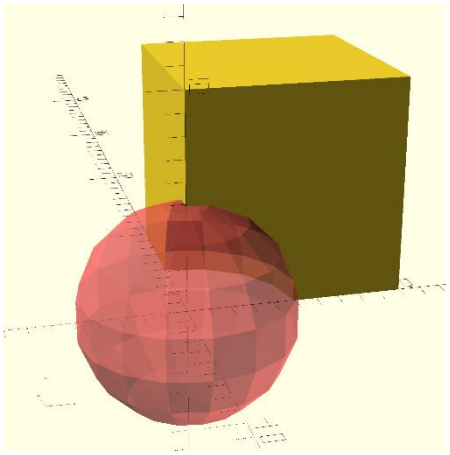
Leicht kann der Überblick verloren gehen, wenn subtrahierte Formen nicht mehr sichtbar sind.
Zur Vereinfachung dieses Fehlers wird ein Rautezeichen (**#**) vor eine subtrahierte Form gesetzt. Somit erscheint eine nur halb sichtbare Version dieser Form = das angebliche „Geisterbild“.

Der nachfolgende Code ist identisch mit dem vorhergehenden Code, außer dass er eine Raute **#** verwendet, um die Kugel als schemenhaftes Bild darzustellen

```
difference() {
cube([10, 10, 10]);
#sphere(5);
}
```

Bild:

Eine Geisterversion einer subtrahierten Kugel, um bei Problemlösungen zu helfen



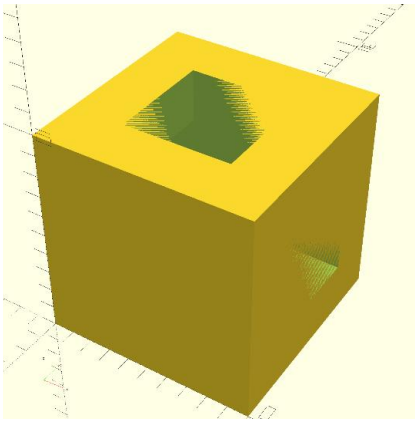
Wichtig:

Die Nutzung der Raute # dient zur Vorschau sowie zur eigenen Kontrolle.

Ist alles in Ordnung, kann anschließend die Raute **#** wieder entfernt werden.

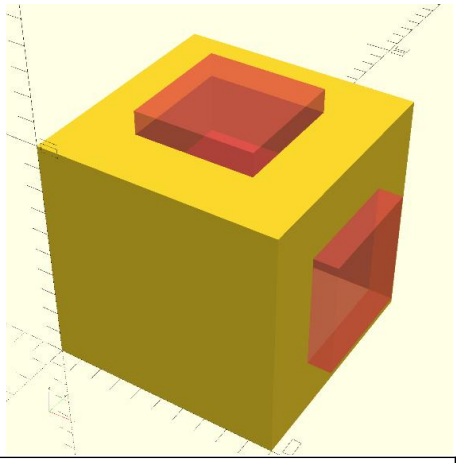
„Schillernde Flächen“ vermeiden bei dem Befehl „difference“

Bild: Zwei Würfel werden von einem größeren abgezogen.



Die schillernden Flächen erscheinen, weil die subtrahierten Formen sich die Fläche mit der Grundform teilen. Somit kommt OpensCAD in Schwierigkeiten – soll die Fläche nun bleiben oder nicht? Aus diesem Grund ist ein Modell mit schillernden Flächen nicht druckfähig.

Lösung:



Die zu subtrahierenden Formen werden in diese Richtung geringfügig vergrößert, damit sie über diese Flächen hinausragen.

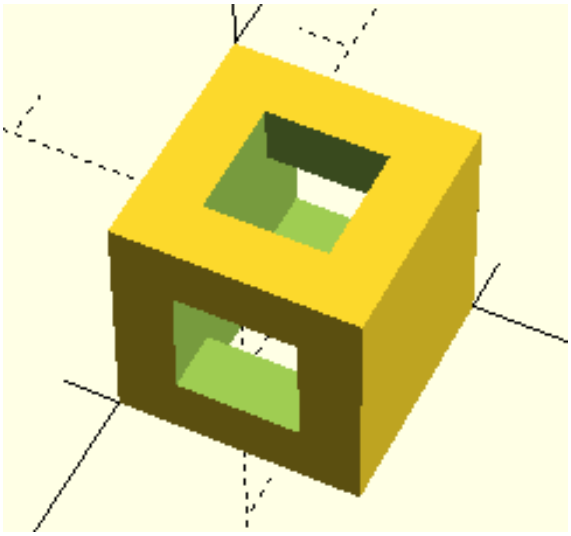
So wird für OpenSCAD definitiv erklärt: ausschneiden!

Im Bild: Von einem großen Würfel werden zwei leicht vergrößerte Würfel subtrahiert.

```
difference() {
cube([10, 10, 10]);
translate([-1, 2.5, 2.5]) cube([12, 5, 5]);
translate([2.5, 2.5, -1]) cube([5, 5, 12]);
}
```

Sobald die „Geisterformen“ entfernt wurden, sollte die verbleibende Form keine schillernden Flächen mehr enthalten.

Eine subtrahierte Form, die jetzt für den 3D-Druck geeignet ist.



004-08

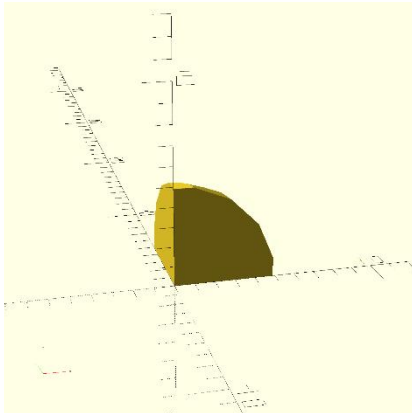
Intersection ( ) { ….. }

Abschneiden überlappender Formen / Schnittmenge

Es kann auch alles weggeschnitten werden bis auf den überlappenden Teil von zwei Formen: Befehls „intersection“

```
intersection() {
sphere(5);
cube([10, 10, 10]);
}
```

Im Bild:
Der Abschnitt einer überlappenden Kugel und einem Würfel, erstellt mit dem Befehl intersection.



Zuerst wird der Befehl intersection gestellt, gefolgt von runden Klammern. Danach kommen geschweifte Klammern, zwischen die mindestens zwei zu erstellende Formen eingetragen werden. Hier ist die Reihenfolge der Formen nicht entscheidend.

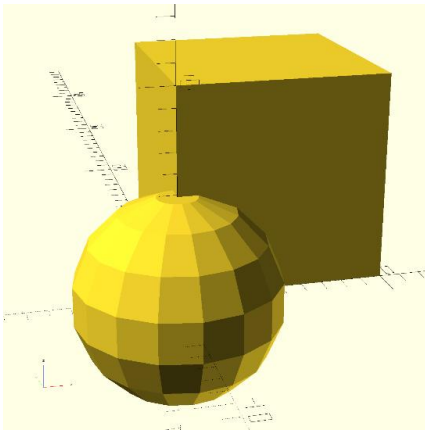
union ( ) { ….. }

Formen verbinden

Die Vereinigungsoperation wird verwendet, um mehrere Formen zu einer einzigen Einheit zu gruppieren.

```
union() {
    cube([10, 10, 10]);
    sphere(5);
}
```

Eine Kugel und ein Würfel wurden über „union“ zu einer Einheit gruppiert.



Der Befehl „**union**“ gruppiert alle Formen innerhalb der geschweiften Klammern zu einer Einheit. Das Einrücken aller Zeilen zwischen den geschweiften Klammern macht den Code lesbarer und dadurch verständlicher.

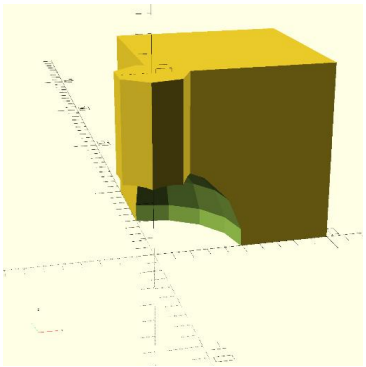
Obwohl es so aussieht, als könnte man Formen durch einfaches Zeichnen kombinieren, wenn diese übereinander liegen, bleibt jedoch jede einzelne Form für sich eine separate Einheit.

Dies kann ein Problem werden, wenn der Differenz-Befehl verwendet werden soll, da dieser Befehl nur von der ersten Form innerhalb der geschweiften Klammern subtrahiert.

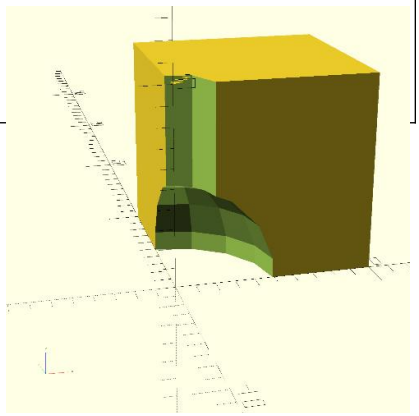
Durch „**union**“ können mehrere Formen zu einer Einheit gruppiert werden, die dadurch anschließend gemeinsam weiter verarbeitet werden können.

```
difference() {
  union() {
        cube([10, 10, 10]);
        cylinder(h=10, r1=2, r2=2);
  }
sphere(5);
}
```

Als Beispiel wird ein Würfel und ein Zylinder mittels „union“ gruppiert. Anschließend wird von ihr die Kugel subtrahiert.



```
difference() {
   cube([10, 10, 10]);
   cylinder(h=10, r1=2, r2=2);
  sphere(5);
}
```



Ohne „union“ würde OpenSCAD stattdessen sowohl den Zylinder als auch die Kugel vom Würfel subtrahieren!