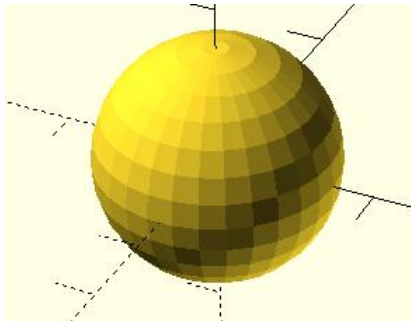
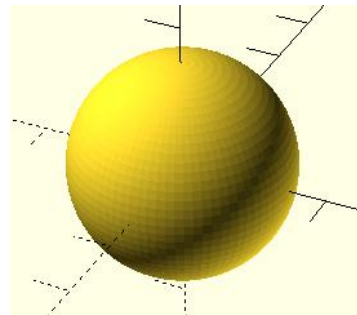


## sphere ( Größe ) ; Zeichnen von Kugeln

sphere ( r=15 ) ;



sphere ( r=15, \$fn=80 ) ;



Um eine Kugel zu zeichnen, wird der Befehl „sphere“ verwendet. Ihm folgt in runden Klammern ( ) der Radius der Kugel, um deren Größe anzugeben.

Am sinnvollsten ist, dass der Radius mit „r“ oder den Durchmesser mit „d“ angegeben wird. Ohne Angabe von „r“ oder „d“ wird immer der Wert als Radius angesehen. Es bedeutet demnach „sphere(10);“ das gleiche wie „sphere(r=10);“ und beides würde in jedem Fall eine Kugel vom Radius 10 erzeugen.

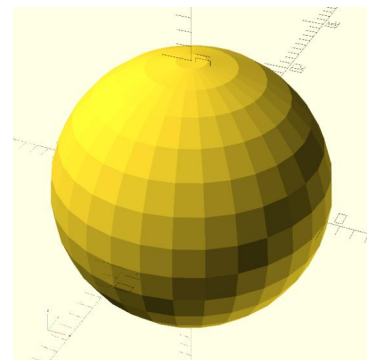
Im folgenden Beispiel wird eine Kugel mit einem Radius von 10 Einheiten gezeichnet:

sphere(10); oder sphere (r=10); oder sphere(d=20);

Eine Kugel mit einem Radius von 10 Einheiten:

Die Größe der Kugel kann geändert werden, indem ihr Radius oder Durchmesser verändert wird. Jedoch nicht wie beim Quader, der drei unterschiedliche Maße hat, besitzt die Kugel die gleichen Maße entlang aller drei Achsen.

Aus diesem Grund enthält der grundlegende Kugelbefehl nur eine Zahl in Klammern. Auch hier wird, wie beim Würfelbefehl, das Ende der Codeanweisung mit einem Semikolon ; beendet.

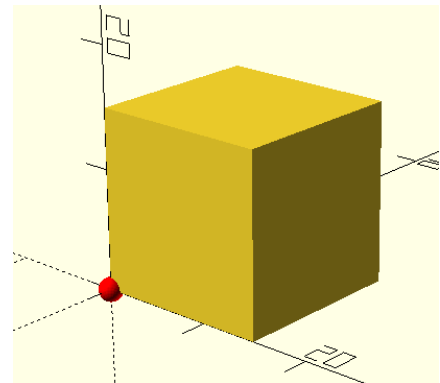


OpenSCAD legt das Zentrum der Kugel immer in den Nullpunkt des Koordinatensystems, also immer in (0, 0, 0).

Ein ,center=false‘ führt nur zu einer Warnung (jedoch kein Fehler!)

## cube ( Größe, center ) ; Erstellen von Würfel

Mit dem Befehl „cube“ wird ein Würfel gezeichnet. Seine Kantenlänge von 15 wird nach dem Befehl in runde Klammern gesetzt, gefolgt von einem Semikolon als Abschluss. Da beim Würfel alle Kanten gleich sind, wird auch nur ein einziger Parameter zur Eingabe benötigt.



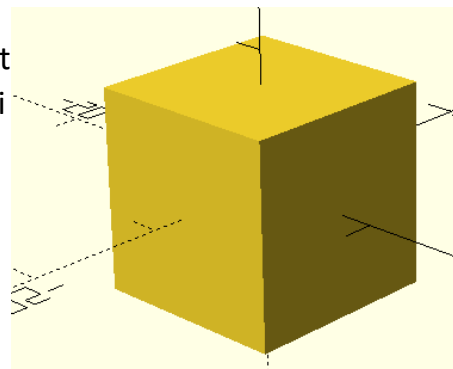
Der Startpunkt zum Zeichnen liegt im Nullpunkt, also (0,0,0). Im Bild mit einem **roten Punkt** dargestellt. Von hier aus wird der Würfel in den positiven Bereich erstellt:

```
cube ( 15 ) ;
```

mit center=true :

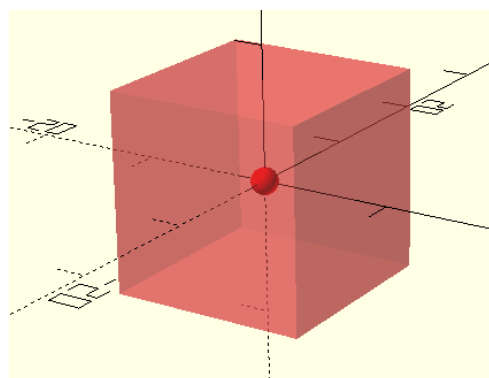
```
cube ( 15, center=true ) ;
```

Wird jedoch „center=true“ mit angeführt, erscheint der Mittelpunkt des Würfels im Nullpunkt aller drei Achsen:



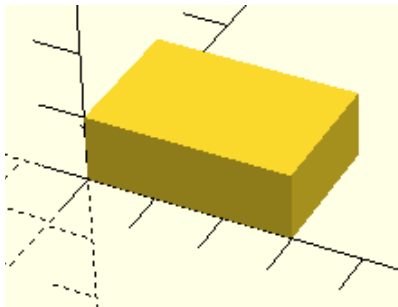
Wird der Würfel transparent gestaltet ( durch #), werden dadurch die einzelnen Achsen – sowie der rote Punkt - in der Mitte des Würfels sichtbar.

```
#cube(15, center=true);  
color("red") sphere(1,$fn=50);
```

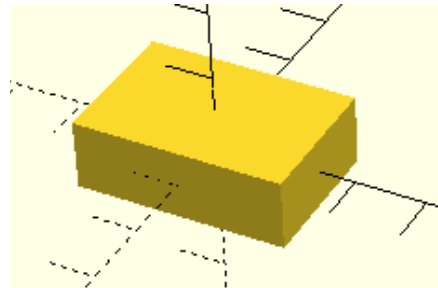


## **cube ( [ Länge, Breite, Höhe ] , center ) ; Erstellen von Quader**

```
cube ( [ 30, 20, 10 ] ) ;
```



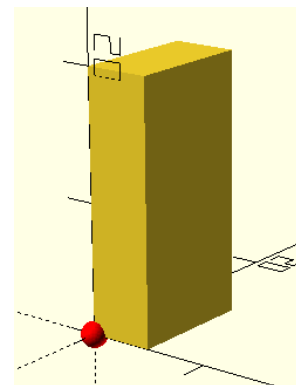
```
cube ( [ 30, 20, 10 ] , center=true) ;
```



Der Befehl „cube“ zeichnet ebenso einen Quader. Dann sind jedoch 3 Parameter notwendig, je einer für X, Y und Z. Auch hier wird im Nullpunkt angefangen zu zeichnen. In den eckigen Klammern [ ] stehen die Werte, die die drei Dimensionen des Quaders darstellen.

Die Reihenfolge der Zahlen-Werte sind wichtig:  
5 ist die Breite des Quader entlang der X-Achse,  
10 ist die Länge des Quaders entlang der Y-Achse,  
und 20 ist die Höhe des Quaders entlang der Z-Achse.

Die gesamte Anweisung wird in runden Klammern ( )  
eingefasst und mit einem Semikolon ; abgeschlossen.

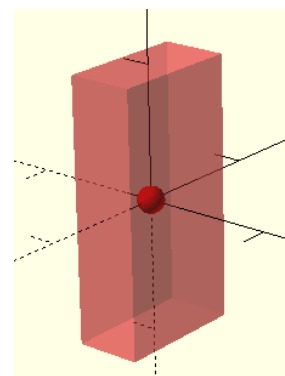


```
cube ( [ 5, 10, 20 ] ) ;
```

```
cube ( [ Wert der X-Achse, Wert der Y-Achse, Wert der Z-Achse ] ) ;
```

Optional kann noch center=true mit angegeben werden.

Durchsichtig dargestellt sind die drei Achsen wieder im Mittelpunkt des Quaders, sowie der rote Punkt im Nullpunkt, sichtbar.



002-04

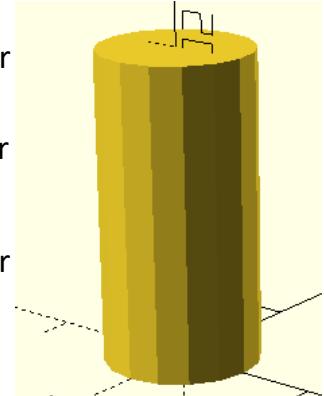
**cylinder ( h = Höhe, r = Radius oder d = Durchmesser ) ;**

### ein Zylinder entsteht

Mit dem Befehl „cylinder“ wird ein gleichmäßiger Zylinder erstellt. Dieser beinhaltet in runden Klammern ( ) den Wert der Höhe, sowie den vom Radius oder Durchmesser.

Die Höhe wird mit h =, der Radius mit r= oder der Durchmesser mit d= angegeben.

Die folgende Anweisung zeichnet einen gleichmäßigen Zylinder dessen Startpunkt im Nullpunkt liegt:

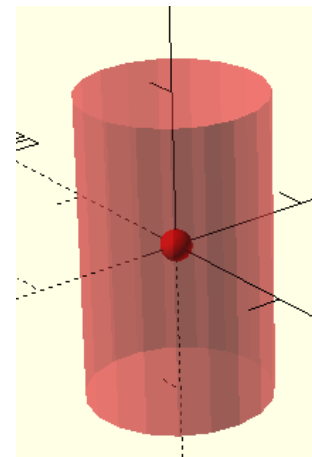


```
cylinder (h=20, r=5); oder cylinder(h=20, d=10);
```

Beim Zylinder sitzt dessen Mittelpunkt des unteren Radius genau auf dem Nullpunkt.

Wird allerdings noch ein „center=true“ angehängt, so erscheint der Mittelpunkt des einfachen Zylinders wiederum im Nullpunkt des Koordinaten-Systems.

Auch hier transparent dargestellt wird der Durchblick auf das Koordinatenkreuz ersichtlich.



```
cylinder(h=20, r=6, center=true);
```

oder

```
cylinder(h=20, d=12, center=true);
```

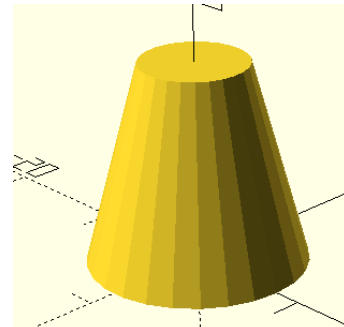
**cylinder ( h=Höhe, r1= Radius 1, r2= Radius 2 ) ;**

002-05

**cylinder ( h=Höhe, d1= Durchmesser 1 , d2= Durchmesser 2 ) ;**

### ein Kegelstumpf entsteht

Die beiden Radien eines Zylinders müssen nicht immer die gleichen Maße haben. Sind diese unterschiedlich, sieht der Zylinder eher aus wie ein Kegel mit abgeschnittener Spitze, wie im Bild gezeigt:



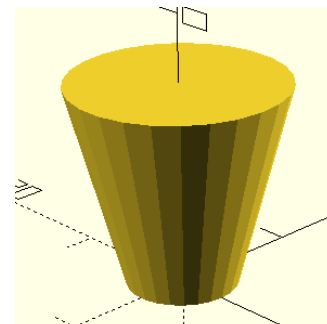
Bei diesem Kegelstumpf wird auch der Befehl „cylinder“ verwendet:

```
cylinder(h=15, r1=8, r2=4); oder cylinder(h=15, d1=16, d2=8);
```

Hier werden nach dem cube-Befehl in Klammern drei Parameter mit Beschriftung gesetzt: **h** für die gesamte Höhe, **r1** für den unteren Radius und **r2** für den oberen Radius. Anstatt **r1** und **r2** kann wiederum **d1** und **d2** verwendet werden. Die drei Werte werden durch Kommata getrennt und zum Schluss mit einem Semikolon versehen.

Beim Kegelstumpf sitzt dessen Mittelpunkt des unteren Radius auch auf dem Nullpunkt.

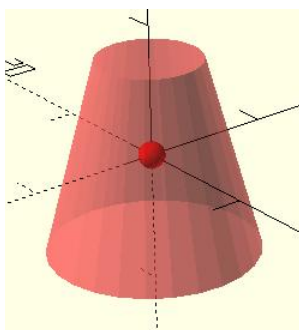
Werden die Werte der Radien / Durchmesser vertauscht, liegt der größere Radius oben und der kleine unten: der Kegelstumpf steht somit auf dem Kopf – genau auf dem Nullpunkt.



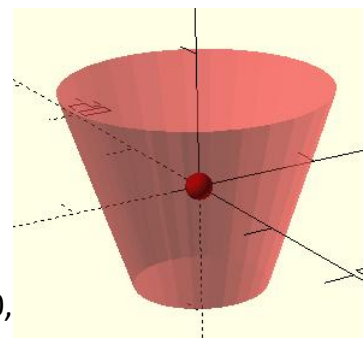
```
cylinder(h=15, r1=4, r2=8); oder cylinder(h=15, d1=8, d2=16);
```

Auch hier gilt „**center=true**“ für den Mittelpunkt im Nullpunkt:

```
cylinder(h= 15, r1= 8, r2= 4,center=true);
```



```
cylinder(h=15, r1=5, r2=10,  
center=true);
```



002-06

**cylinder ( h= Höhe, r1= Radius 1, r2= 0 ) ;**

**cylinder ( h= Höhe, d1= Durchmesser 1 , d2= 0 ) ;**

**einen Kegel zeichnen**

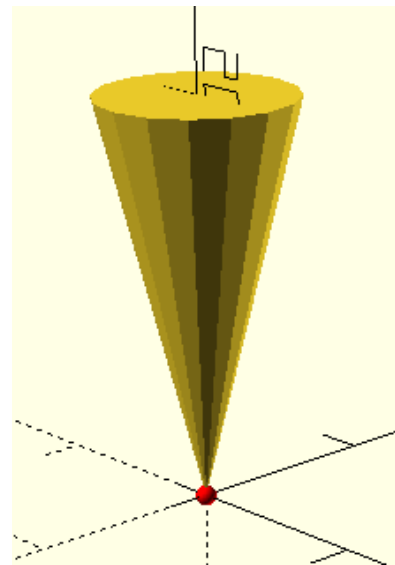
Es kann ein spitzer Kegel gezeichnet werden (siehe Bild), indem einem der Radien ein Wert von 0 zugewiesen wird.

```
cylinder(h=20, r1=0, r2=5); oder cylinder(h=20, d1=0, d2=10);
```

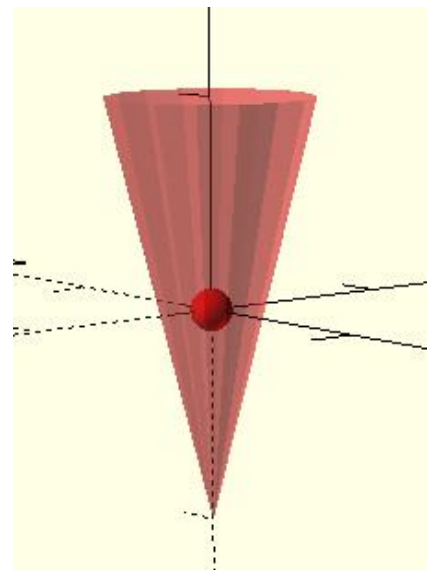
Ein Kegel auf der Spitze stehend mit einer Höhe von 20, einem unteren Radius r1 von 0 und einem oberen Radius r2 von 5.

Soll der Kegel nach oben zeigen, sind die Werte muss der obere Radius r2 Null sein.

Ebenso beim Kegel sitzt dessen Mittelpunkt des unteren Radius genau auf dem Nullpunkt.

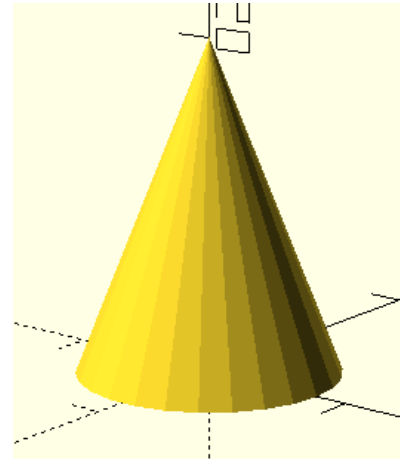


Wird „**center=true**“ angefügt, so erscheint der Mittelpunkt des Kegel wiederum im Nullpunkt des Koordinatensystems.



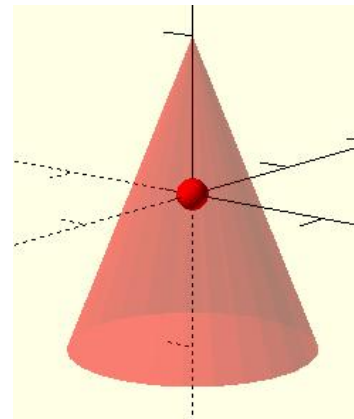
```
cylinder(h=20, r1=0, r2=5, center=true);
```

Für einen Kegel bei dem die Spitze nach oben zeigen soll, werden nur die zwei Radien geändert: unten der größere und oben gleich Null.



```
cylinder(h=20, r1=8, r2=0);
```

Wird „**center=true**“ angefügt, so erscheint der Mittelpunkt des Kegel wiederum im Nullpunkt des Koordinatensystems.



```
cylinder(h=20, r1=8, r2=0; center=true);
```

## center=true: Formen zentrieren

Wenn ein Design mit mehreren Formen erstellt wird, sollte man wissen, wie die Formen im Vorschauenfenster bewegt werden können. Ansonsten, wenn alle standardmäßig aufeinander sitzen, überschneiden sie sich.

Als Beispiel soll folgendes Muster dienen:

```
cube([20, 10, 10]);  
sphere(5);  
cylinder(h=30, r=2);
```

Im Bild: Mehrere Grundformen mit der standardmäßigen Vorgabe

Zentrieren von Formen mit dem Befehlszusatz: „center=true“.

Standardmäßig zeichnet der Befehl „sphere“ eine Kugel so, dass ihr Mittelpunkt im Nullpunkt liegt; beim cube-, cylinder- und import- Befehl geschieht dies jedoch nicht.

Sollten andere Formen so gezeichnet werden, dass sie ebenfalls um den Nullpunkt zentriert sind, muss als Parameter center=true zwischen den Klammern ], }; eingefügt werden, wie im folgenden Beispiel:

```
Cube ( [ 5, 10, 20 ] , center=true ) ;
```

Im Bild:

Nun liegt der Mittelpunkt des Quaders bei (0, 0, 0).

So können Zylinderformen und Kegel durch die Erweiterung des Parameters von „center=true“ in den Nullpunkt gestellt werden.

Bei importierten Formen ist es **nicht möglich** „center=true“ zu verwenden!

