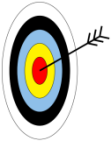
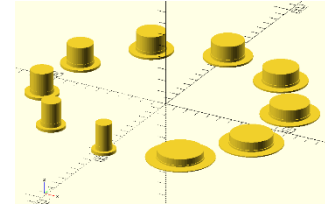


Modules



Hier wird euch erklärt, wie ihr eigene Module benutzen könntet, um euch schnell Objekte zu kopieren oder untereinander zu teilen.



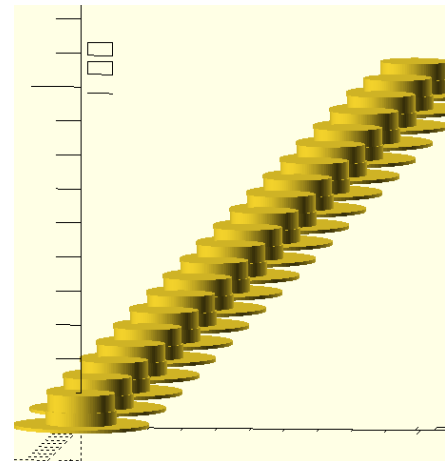
Ein eigener Befehl

Mit „`module name()`“ könnt ihr einen eigenen Befehl erstellen. Immer wenn ihr irgendwo in eurem Programm den Namen des Befehls verwendet, wird alles in den geschweiften Klammern eures Befehls ausgeführt. So könnt ihr leicht eigene Objekte vervielfältigen:

```

module hut() {
  cylinder(r=20,h=1,$fn=100);
  cylinder(r=10.5,h=5,$fn=100);
  cylinder(r=10,h=10,$fn=100);
}

for (i = [0:20]) {
  translate([0,i*50,0]) {
    hut();
  }
}
  
```



Die Parameter

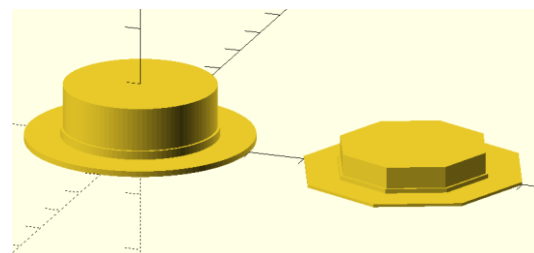
Ihr könnt eigene Parameter für euren Befehl definieren, indem ihr sie einfach hinter dem Namen in die Klammern schreibt. Ihr könnt sie dann innerhalb der geschweiften Klammern benutzen, um unterschiedliche Objekte zu erzeugen:

```

module hut(radius,hoehe,feinheit) {
  cylinder(r=radius,h=hoehe/10,$fn=feinheit);
  cylinder(r=radius/1.45,h=hoehe/4,$fn=feinheit);
  cylinder(r=radius/1.5,h=hoehe,$fn=feinheit);
}

hut(20,10,100);

translate([50,0,0]) {
  hut(20,5,8);
}
  
```



Modules

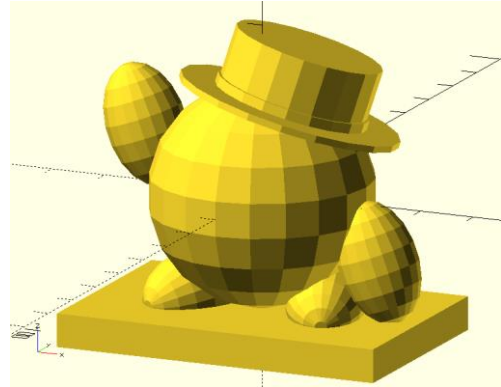
Module teilen

Ihr könnt Module aus anderen Programmen oder Dateien benutzen, indem ihr `use<datei.scad>` benutzt. Dafür müssen alle Dateien im gleichen Ordner sein und ihr müsst ihre Namen kennen:

```
use<Hut.scad>
use<Info_Dot.scad>

infodot();

rotate([-10,15,0]){
  translate([0,0,15]){
    hut(20,12,20);
  }
}
```



Die Dateien müssen alle im gleichen Ordner sein. Der Name muss mit Endung und Groß- und Kleinschreibung richtig geschrieben werden.