

3Dプリンタ実習

2017-1-19 14:00- 11講義室

異分野基礎科学研究所 松本 正和

目標

- 3Dプリンタの動作原理と製作過程を体験する
- 実際に使えるものを作ってみる

製作手順

1. モデルの造形; CADソフト
STLファイルの生成
2. 3Dプリンタで出力

ほかに、3Dスキャナでデータ生成→出力という手順もある

1. モデルの造形

- 形を“練る”。ワープロソフトに相当
- 2種類のCAD (Computer-Aided Design)ソフト
 - インタラクティブ (Blender, SketchUp, etc.)
 - パラメトリック (OpenSCAD, etc.)
- STLファイルを生成 (.docファイルに相当)

2. 3Dプリンタで出力

- 3Dプリンタ (プリンタに相当)
(今回はMakerBot Replicator 2X)
- 制御ソフトウェア (プリンタドライバーに相当)
(今回はMakerBot Desktop)
- 3Dプリンタにはいろんな種類があり、安いものは2万円ぐら
いから、高いものでは数千万円まで。
(試作用か、製品用か、高精度か、特殊素材か、速いか)



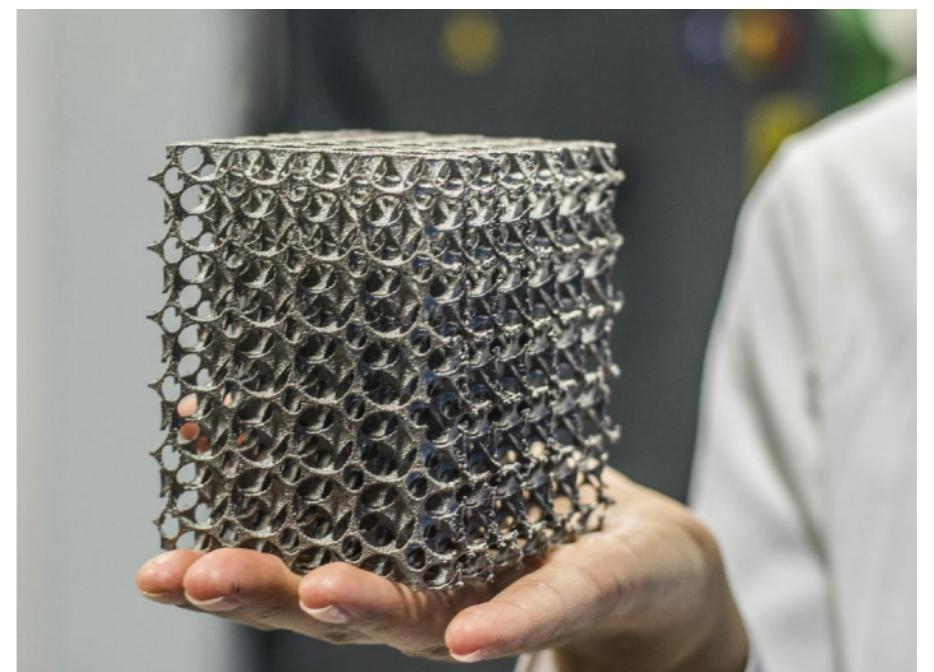
ダヴィンチJr. 1.0(再生品)
プラスチック溶融積層型

¥19,800



3D Systems ProX DMP 320
金属粉体焼結型

~¥20,000,000



3Dプリンタの種類

	速い	強い	軽い	正確	安い
溶融積層型	融かしたプラスチックを少しずつ積み重ねる		○	○	○
光造形型	光硬化樹脂を1層ずつ紫外線で固める	○		○	○
粉体接着型	石膏や金属の粉体をレーザーや接着剤で接着	○			○
シート積層型	シート状材料を断面の形に切って積層する	○	○		○
(切削型)	固まりの素材から刃物で彫刻して造形		○		○

3Dプリンタの概要

- Makerbot社 Replicator 2X
- 融解積層型プリンタ
- ABS / PLA / HIPS / TPUに対応, 2種類を同時利用可
- 最小レイヤー高さ 0.1 mm (通常は0.2 mm)
- 最大造形サイズ 24.6 cm L × 15.2 cm W × 15.5 cm H

溶融積層用の素材

熱で融ける(熱可塑性)プラスチック

いずれも約5,000円 / kg

ABS

ヘッドホンやパソコンなど多くの工業製品に使われる、
固く割れにくい素材。ただし熱収縮が大きいのが難点

PLA

生物由来の素材。ABS並みの強度があり、収縮しないが、
しなやかさに欠け、割れやすい。

HIPS

PLAに似ている。溶剤で融かせるのでABSのサポート材に使える。

TPU

柔軟性があり、非常に強い。スマホのケースに使われる。

新しい素材が続々発表されている。(金属粉や木粉を含む
もの、ナイロン、PET、チョコレート、飴、などなど)

3Dプリンタでできること, できないこと

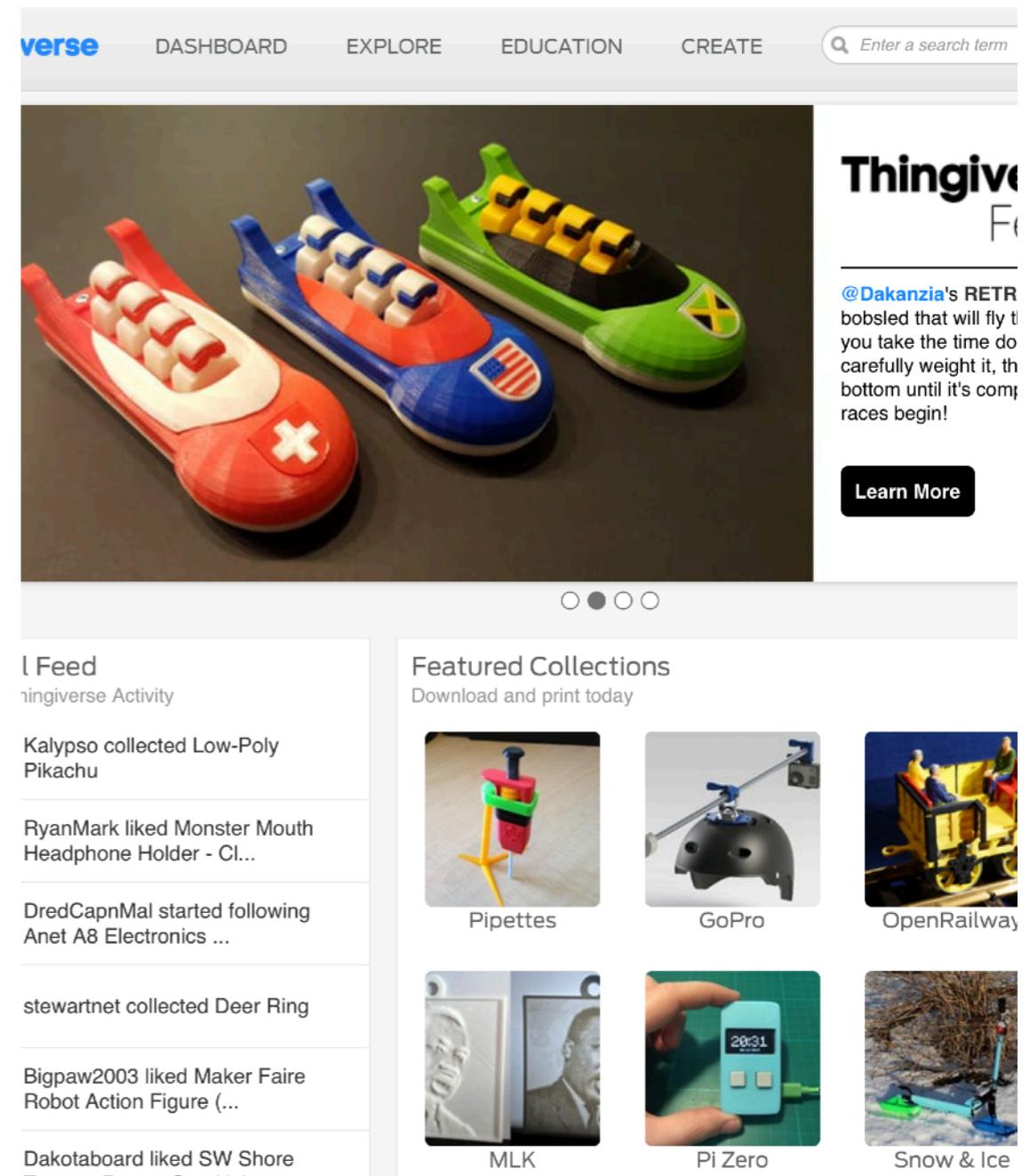
- はじめから噛みあった状態で部品を作れる!
- 軽くて強いものが作れる!
- 大きな物体はとても製作に時間がかかる
- 高架は苦手 (自動サポート生成機能あり)

サポート

- 3Dプリンタといえども、宙に物体を作ることはできない。→向きを工夫する。
- 45度以上のオーバーハングには自動的にサポート(足場)が作られる。
- 手が入らないような複雑な形の場合には、別の素材でサポートを作り、溶剤で溶かして除去する。

OpenSCAD

- プログラム言語. プリミティブと呼ばれる基本図形の和や差(演算)で複雑な形状を表現する.
- 3D模型情報の共有サイト Thingiverse の標準言語
<http://thingiverse.com>

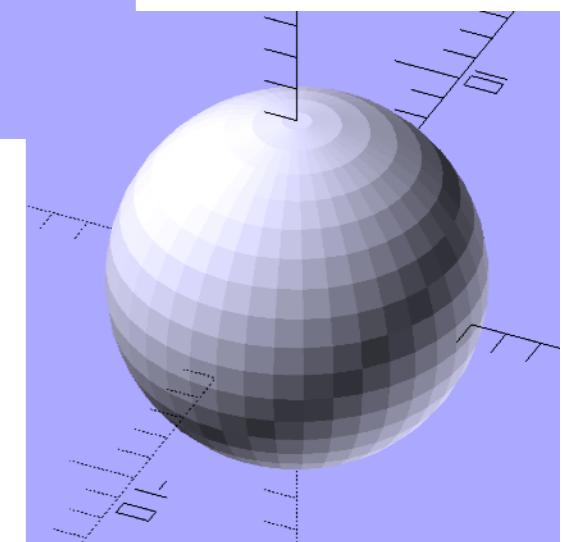
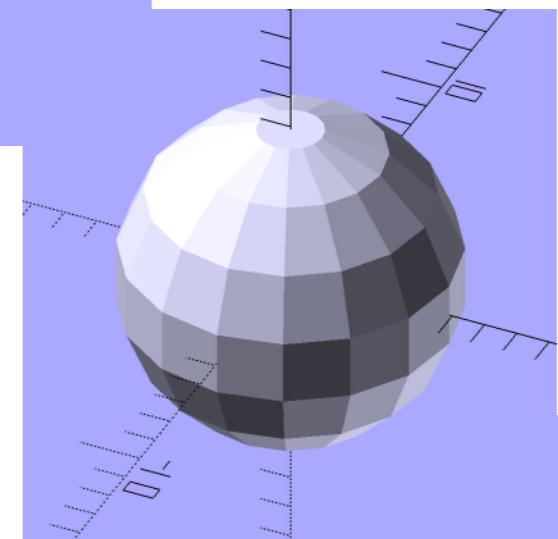
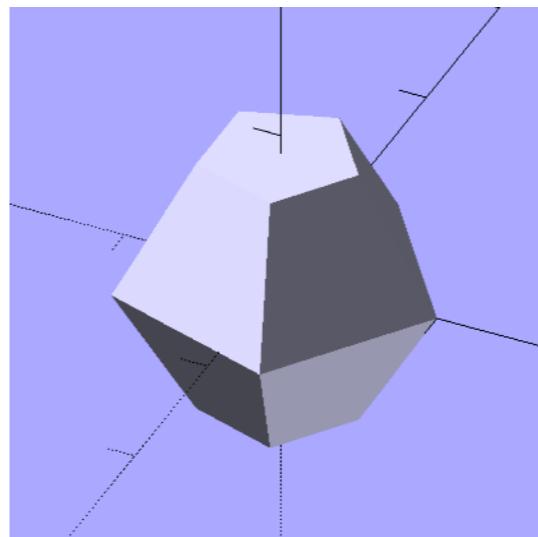


主なプリミティブと演算

球	平行移動	回転
円柱・角柱・円錐・角錐	拡大縮小	鏡映
直方体	繰り返し	条件分岐
多面体	和, 差, 交差	押し出し
	凸包	ミンコフスキーや

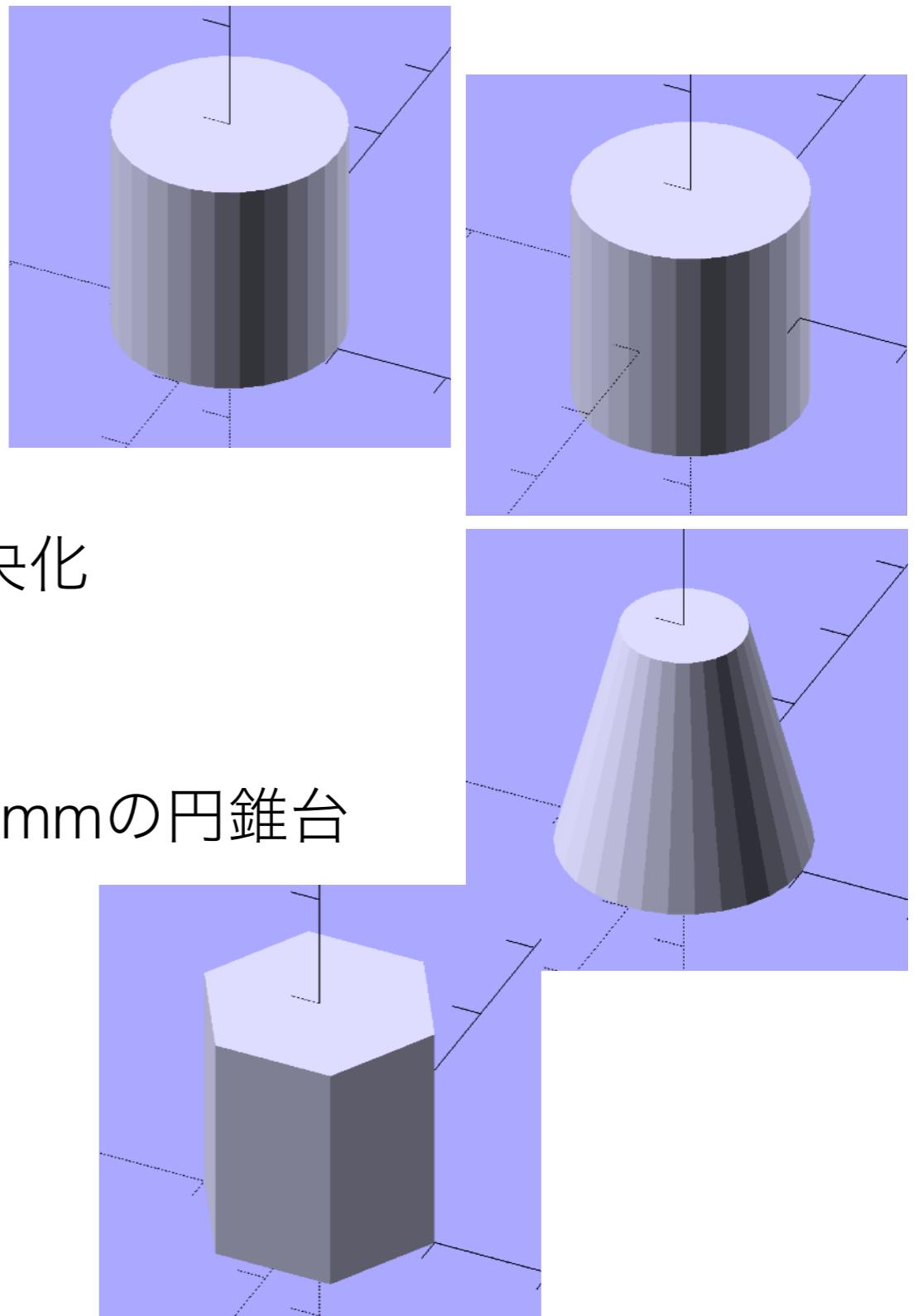
球

- `sphere();` 半径1mmの球
- `sphere(r=5);` 半径5mmの球
- `sphere(r=5, $fn=40);` スムーズな球



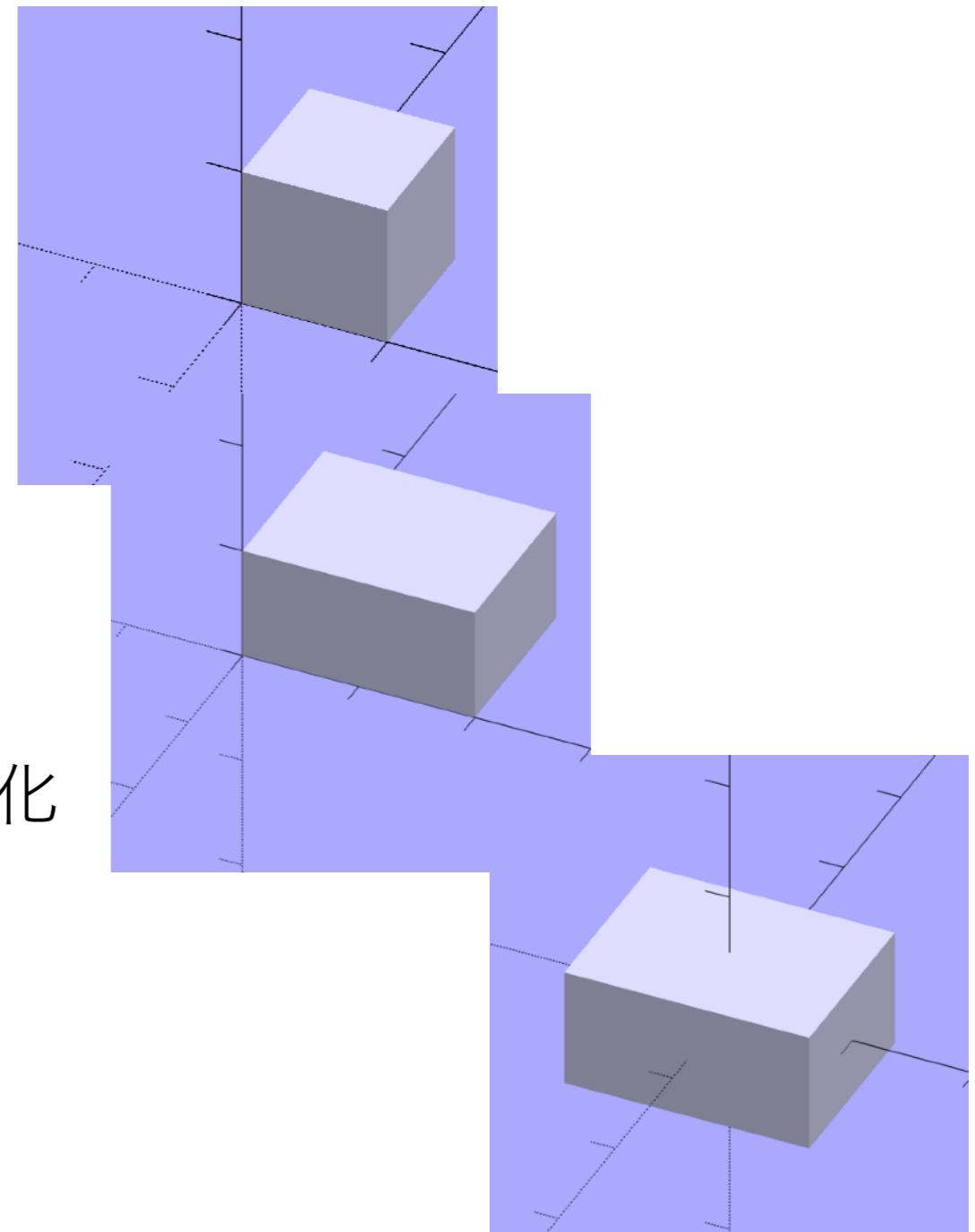
柱

- `cylinder(r=10,h=20);`
半径10mm, 高さ20mmの円柱
- `cylinder(r=10,h=20, center=true);`
半径10mm, 高さ20mmの円柱, 中央化
- `cylinder(r1=10, r2=5, h=20);`
底半径10mm, 頂半径5mm, 高さ20mmの円錐台
- `cylinder(r=10, h=20, $fn=6);`
半径10mm, 高さ20mmの六角柱



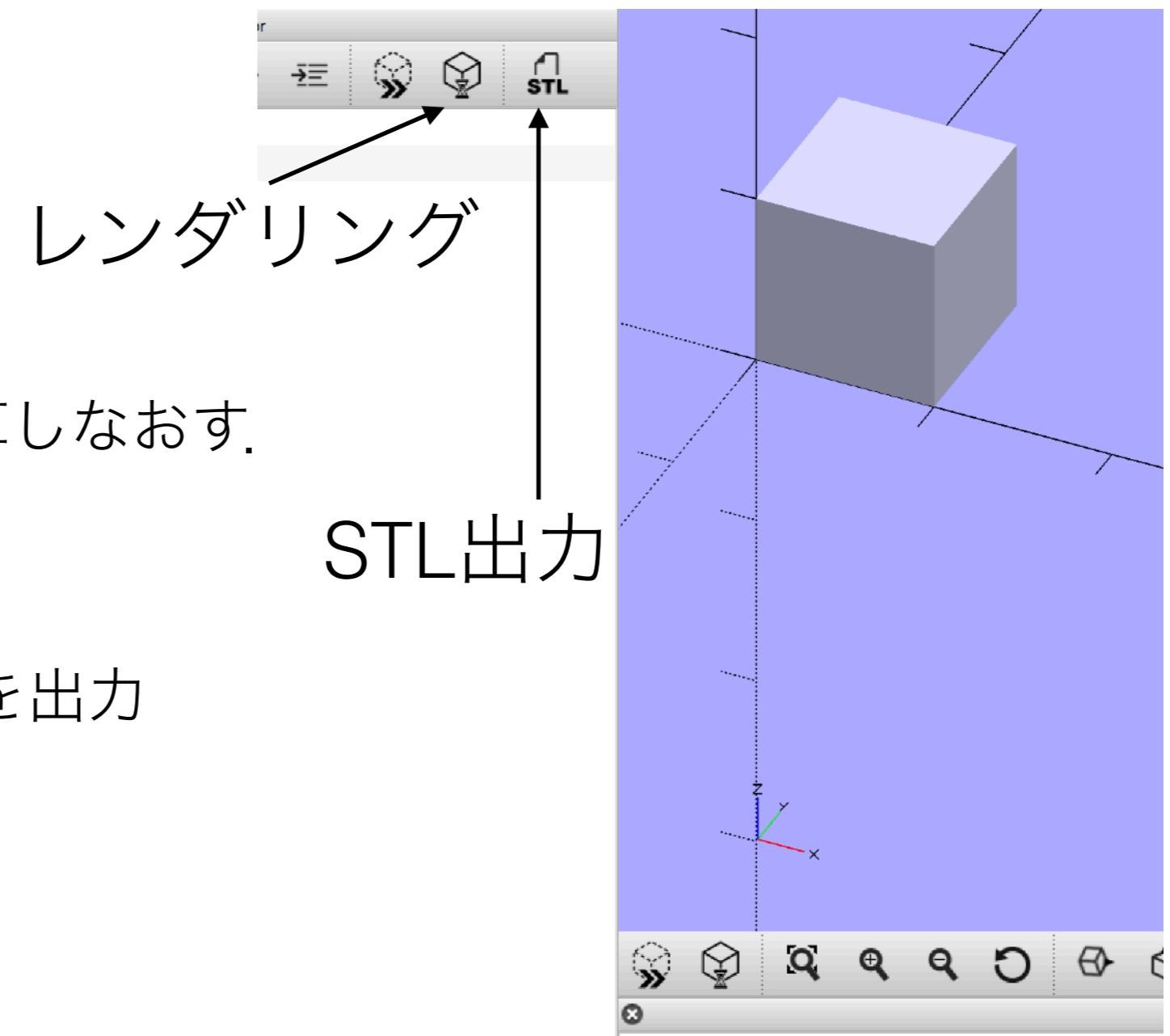
箱

- `cube();` 一邊1mmの立方体
- `cube([20,15,10]);` 直方体
- `cube([20, 15, 10], center=true);` 中央化



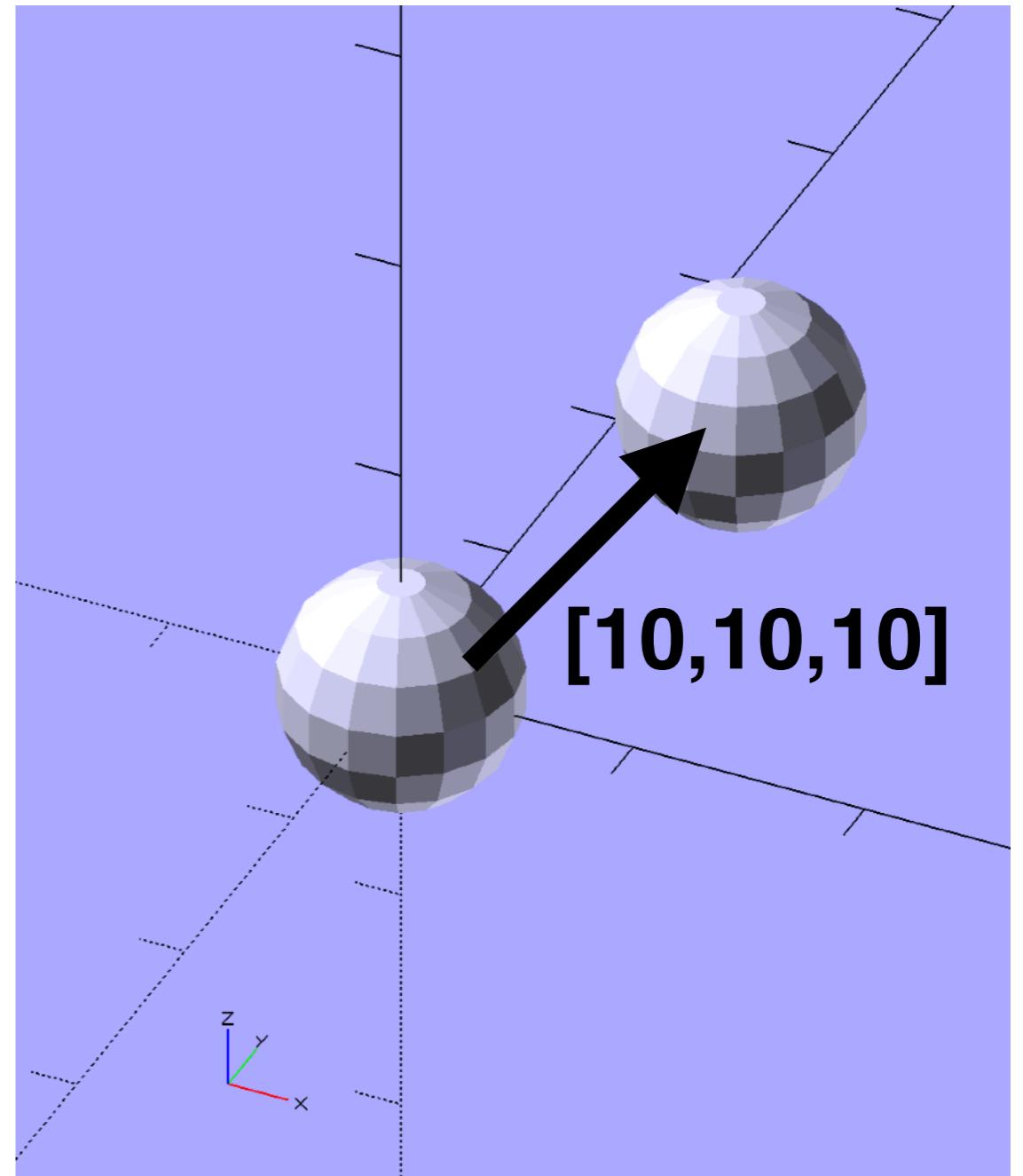
レンダリングとSTL出力

- レンダリング
立体構造をきちんと計算しなおす。
- STL出力
3Dプリンタ用のデータを出力



平行移動・回転・拡大・鏡映

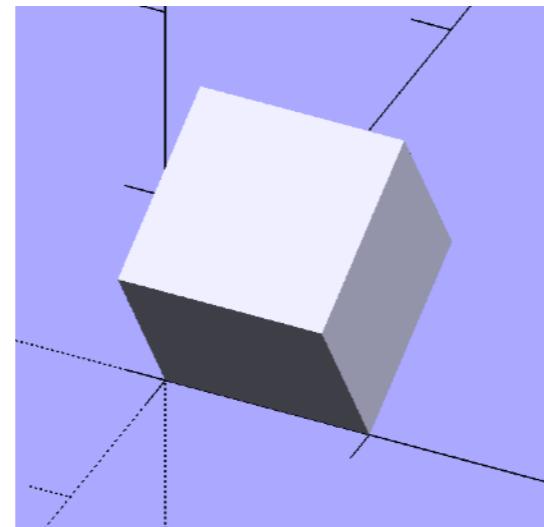
- プリミティブを中括弧{}でくくり、その前にtranslate(並進)やrotate(回転)などを書き加える。
- **translate([10,10,10])**{
 sphere(r=5);
}



平行移動・回転・拡大・鏡映

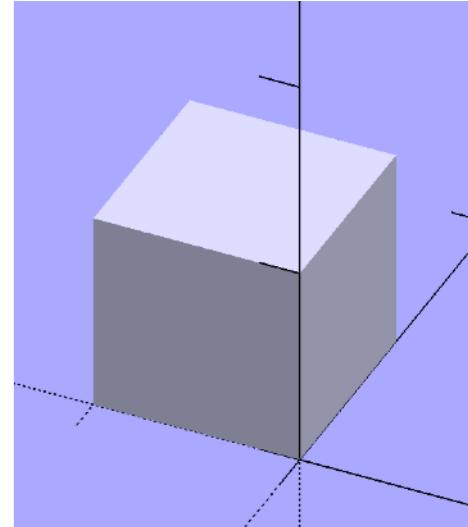
- x軸の周りに30度回転

```
rotate([30,0,0]){
    cube([10,10,10]);
}
```



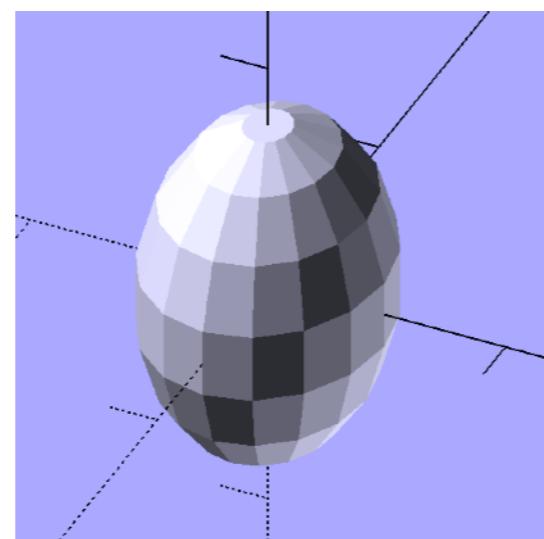
- x軸方向に鏡映

```
mirror([1,0,0]){
    cube([10,10,10]);
}
```



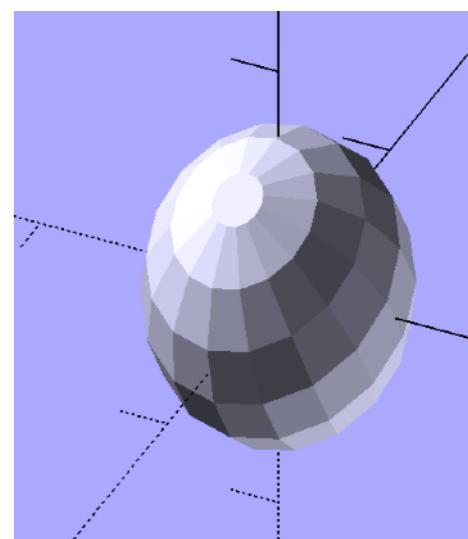
- y, z方向に拡大

```
scale([1,1.2,1.5]){
    sphere(r=5);
}
```



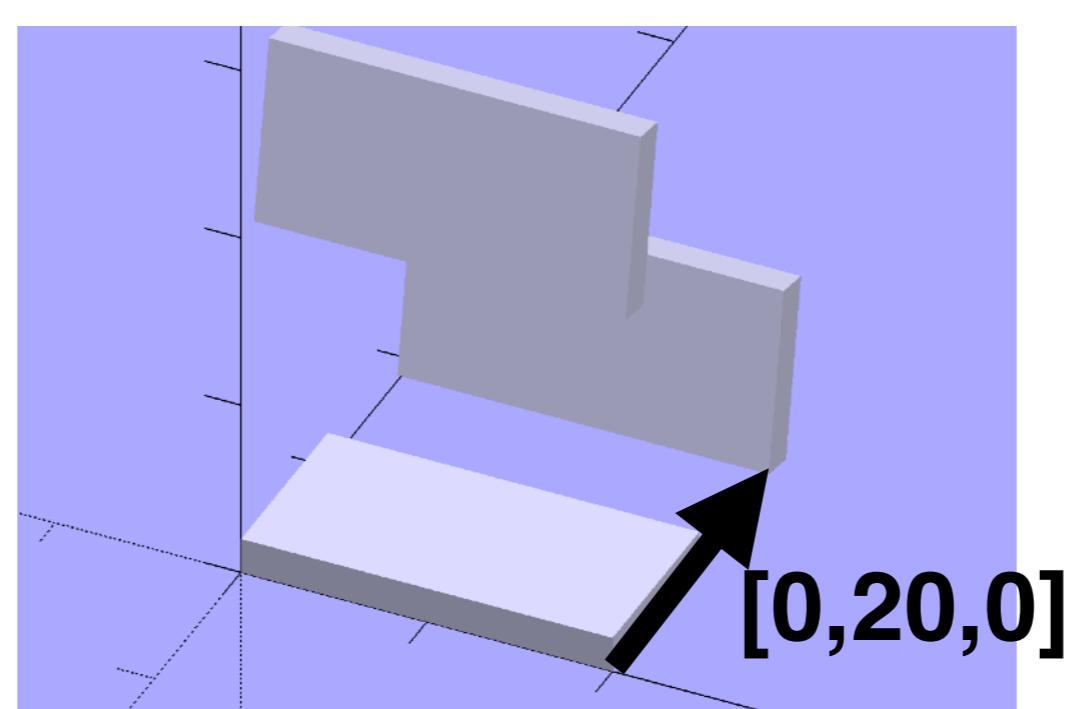
- y, z方向に拡大してから
x軸の周りに30度回転

```
rotate([30,0,0]){
    scale([1,1.2,1.5]){
        sphere(r=5);
    }
}
```



順番が大事

- y方向に20 mm移動してから
x軸の周りに80度回転
- **rotate([80,0,0]) {
translate([0,20,0]) {
cube([20,10,2]);
}
}**
- x軸の周りに80度回転してから
y方向に20 mm移動
- **translate([0,20,0]) {
rotate([80,0,0]) {
cube([20,10,2]);
}
}**

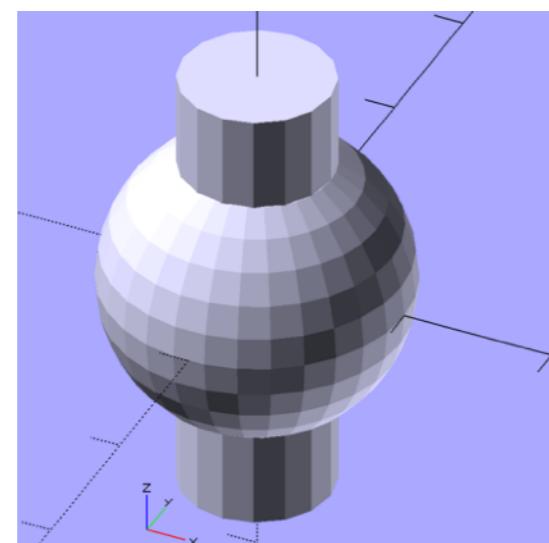


集合演算(和・差・交差)

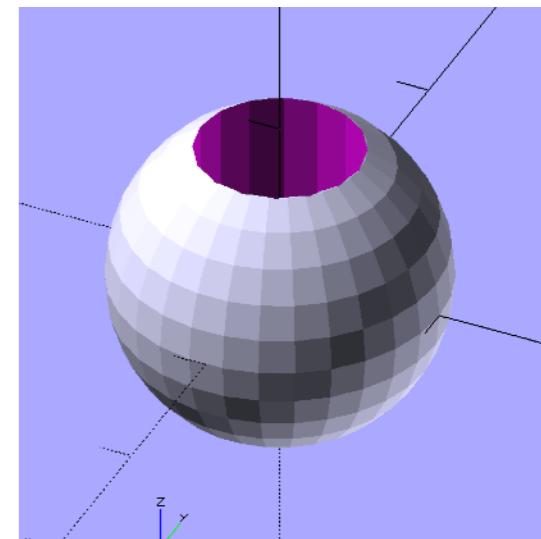
- 和は並べるだけ。あるいは**union**をつかう。

```
sphere(r=10);
cylinder(r=5, h=30, center=true);
または
```

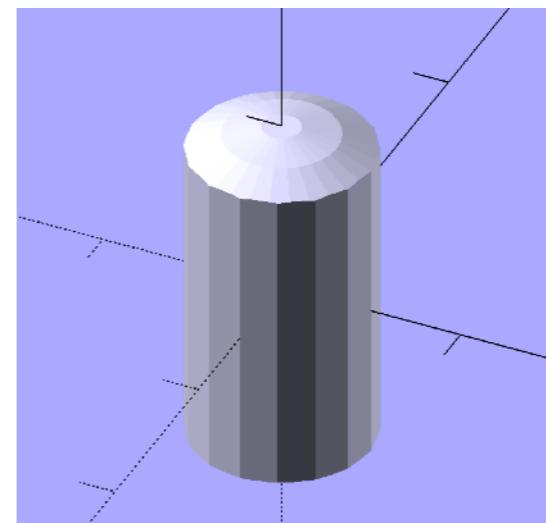
```
union(){
    sphere(r=10);
    cylinder(r=5, h=30, center=true);
}
```



```
difference(){
    sphere(r=10);
    cylinder(r=5, h=30, center=true);
}
```

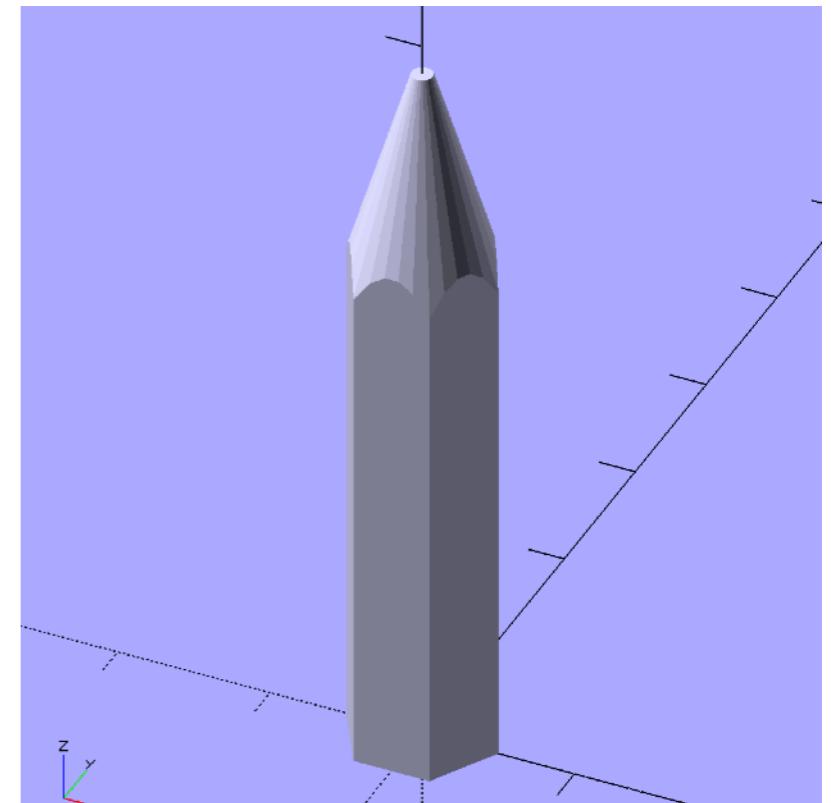


```
intersection(){
    sphere(r=10);
    cylinder(r=5, h=30, center=true);
}
```



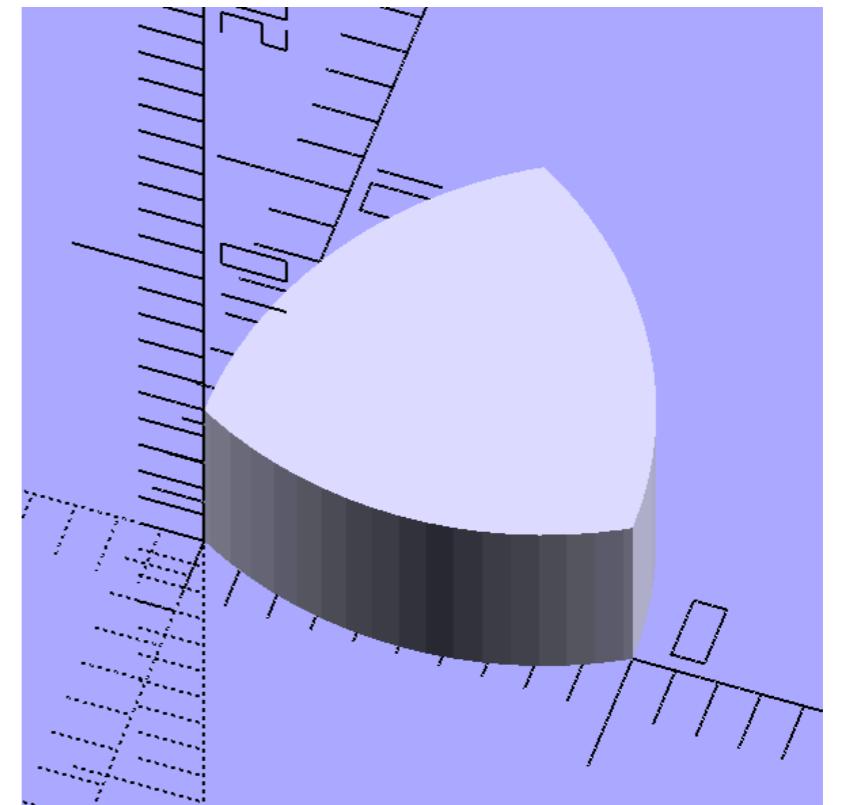
いろいろ組みあわせてみる

```
//6角柱と円錐の交差  
intersection(){  
    cylinder(r=5,h=50,$fn=6);  
    cylinder(r1=15,r2=0.7,h=48);  
}
```



いろいろ組みあわせてみる

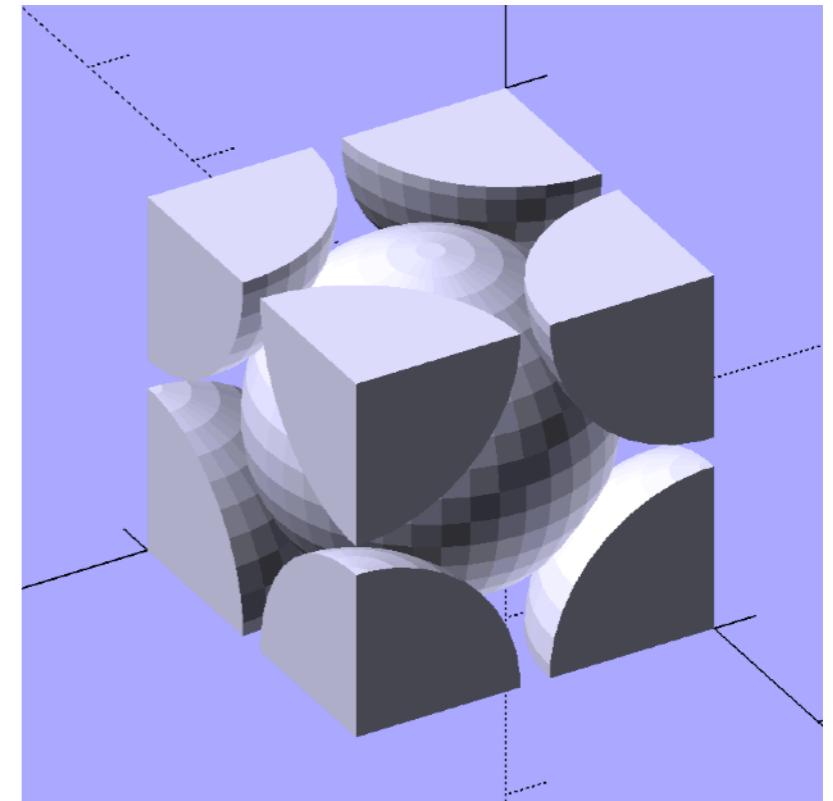
```
//3つの円柱の交差=ルーローの三角柱  
$fn=100;  
intersection(){  
    cylinder(r=10,h=5);  
    translate([10,0,0]){  
        cylinder(r=10,h=5);  
    }  
    translate([5,5*sqrt(3),0]){  
        cylinder(r=10,h=5);  
    }  
}
```



いろいろ組みあわせてみる

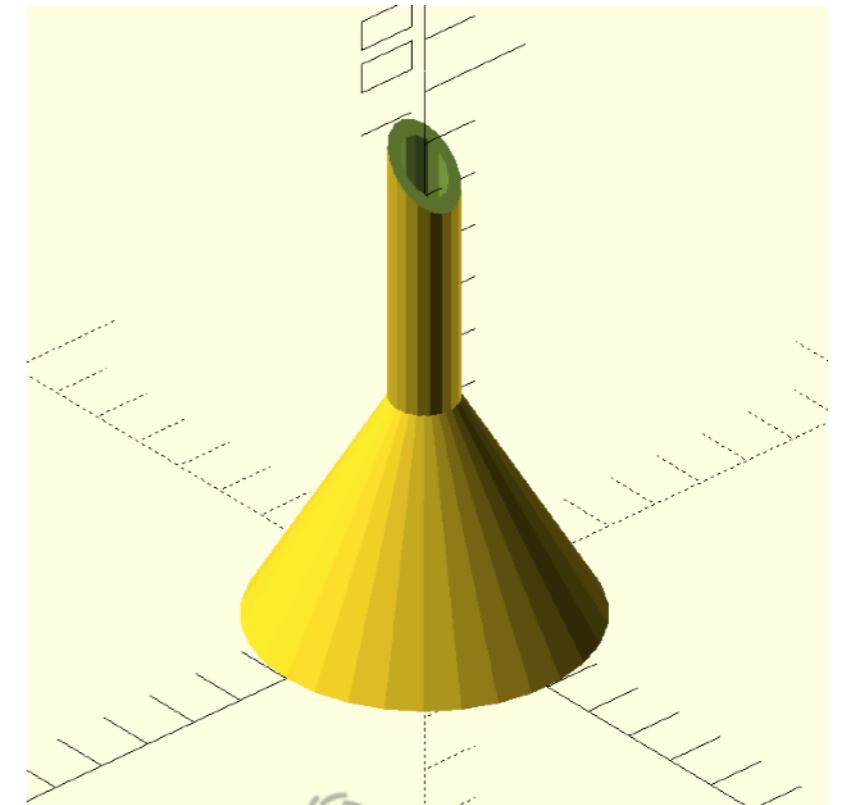
//9つの球と、1つの立方体の交差=BCC

```
$fn=40;  
R = 0.92;  
intersection(){  
    union(){  
        sphere(r=R);  
        translate([1,1,1]){sphere(r=R);}  
        translate([2,0,0]){sphere(r=R);}  
        translate([2,2,0]){sphere(r=R);}  
        translate([0,2,0]){sphere(r=R);}  
        translate([2,0,2]){sphere(r=R);}  
        translate([2,2,2]){sphere(r=R);}  
        translate([0,2,2]){sphere(r=R);}  
        translate([0,0,2]){sphere(r=R);}  
    }  
    cube([2,2,2]);  
}
```



いろいろ組みあわせてみる

```
R=30;  
H=sqrt(3)*R;  
thick=2;  
Rpipe=4;  
  
difference(){  
    union(){  
        cylinder(r1=R, r2=0, h=H);  
        cylinder(r=Rpipe+thick, h=H*2);  
    }  
    translate([0,0,-thick*2]){  
        cylinder(r1=R, r2=0, h=H);  
    }  
    cylinder(r=Rpipe, h=H*2+1);  
    translate([0,6,H*2-3]){  
        rotate([45,0,0]){  
            cube([30,30,30],center=true);  
        }  
    }  
}
```



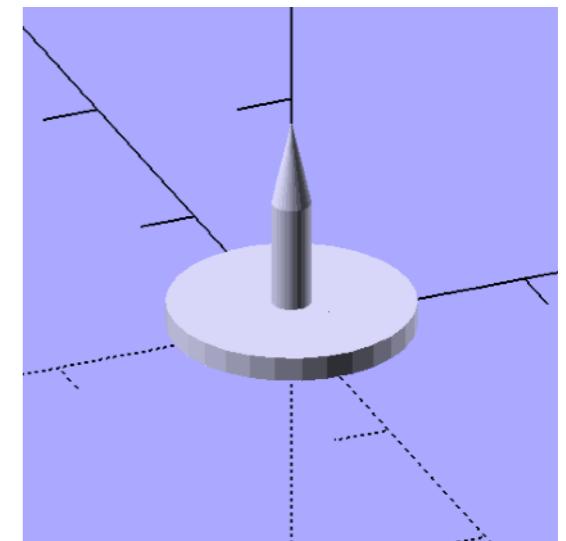
変数

- 変数を使うと、数値の意味がわかりやすくなり、あとでまとめて変更しやすくなる。

```
$fn=30;  
cylinder(r=5,h=1);  
intersection(){  
    cylinder(r=0.8,h=9);  
    cylinder(r1=2,r2=0,h=9);  
}
```



```
$fn=30;  
head_radius=5;  
head_thick=1;  
pin_length=9;  
pin_radius=0.8;  
pin_base=2;  
cylinder(r=head_radius,h=head_thick);  
intersection(){  
    cylinder(r=pin_radius,h=pin_length);  
    cylinder(r1=pin_base,r2=0,h=pin_length);  
}
```



条件分け

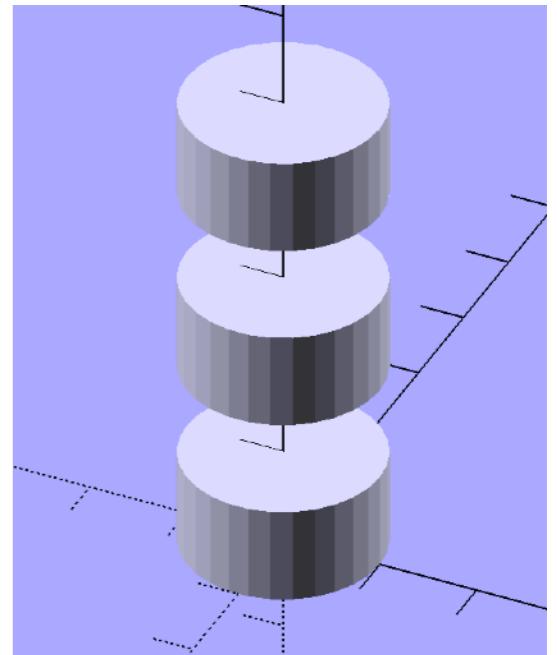
- 通常のプログラム言語同様,
if文で条件分岐できます。
書き方はC言語とほぼ同じ。
(ただし、あまり使う機会はありません。)

```
if ( x < 10 ){
    cylinder(r=10,h=20);
}else{
    sphere(r=10);
}
```

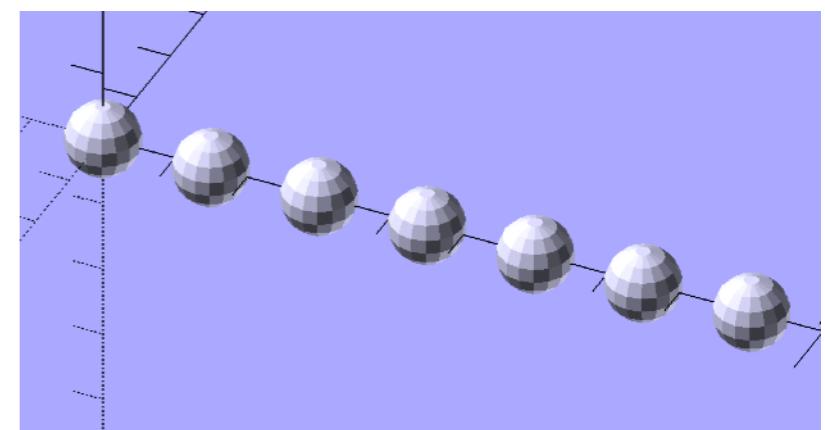
繰り返し

- 繰り返し構文もあります。
かなり独特の書き方なので、
慣れが要るかも。

```
for ( z=[0,20,40] ){
    translate([0,0,z]){
        cylinder(r=10,h=10);
    }
} //zは0, 20, 40
```



```
for ( a=[0:15:100] ){
    translate([a,0,0]){
        sphere(r=5,h=10);
    }
} //aは0から15間隔で100まで
```

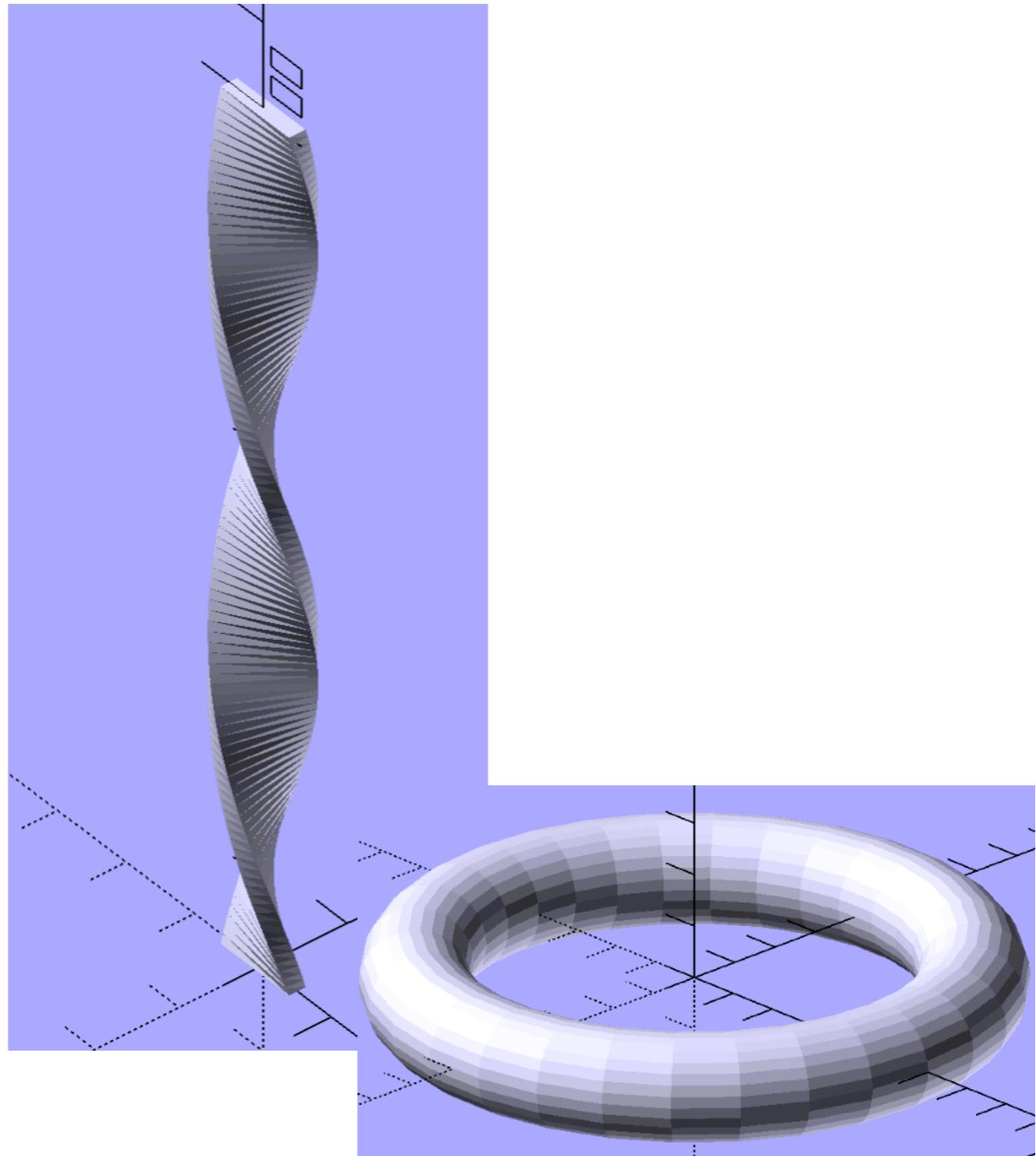


面白い機能1：押し出し

- 平面図形(squareまたはcircle)を押し出して立体にする。まっすぐ押す, らせんを描く, 回転しながら押す, などがある。

```
linear_extrude(height=100,  
                twist=360,  
                slices=100){  
    square([10,2],center=true);  
}
```

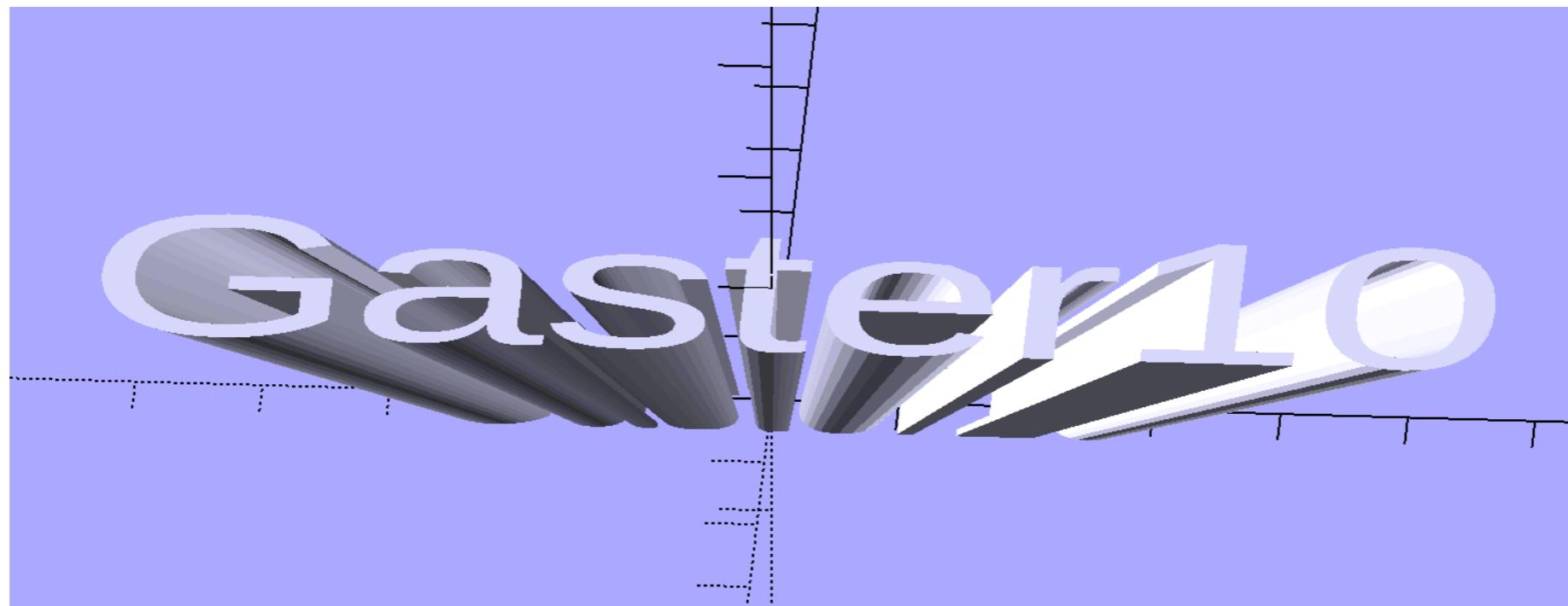
```
rotate_extrude(slices=100){  
    translate([50,0,0])  
    circle(r=10);  
}
```



面白い機能2：文字

- 押し出しを利用して、立体文字も作れます。
(今のところアルファベットのみ)

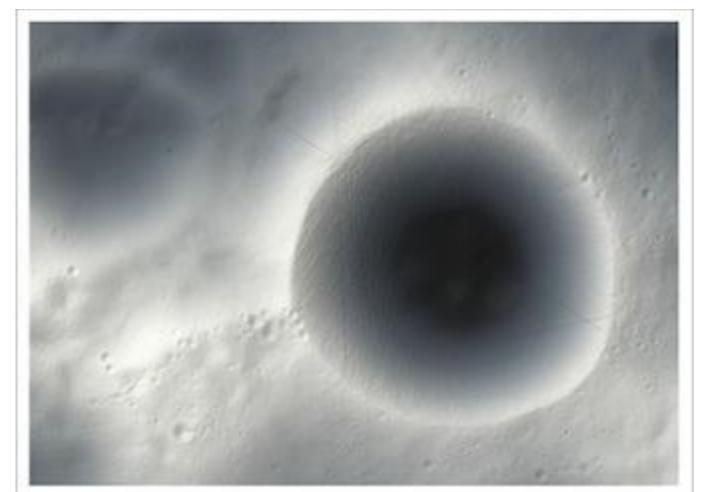
```
linear_extrude(height=10,scale=2){  
    translate([-27,-5,0]){  
        text("Gaster10");  
    }  
}
```



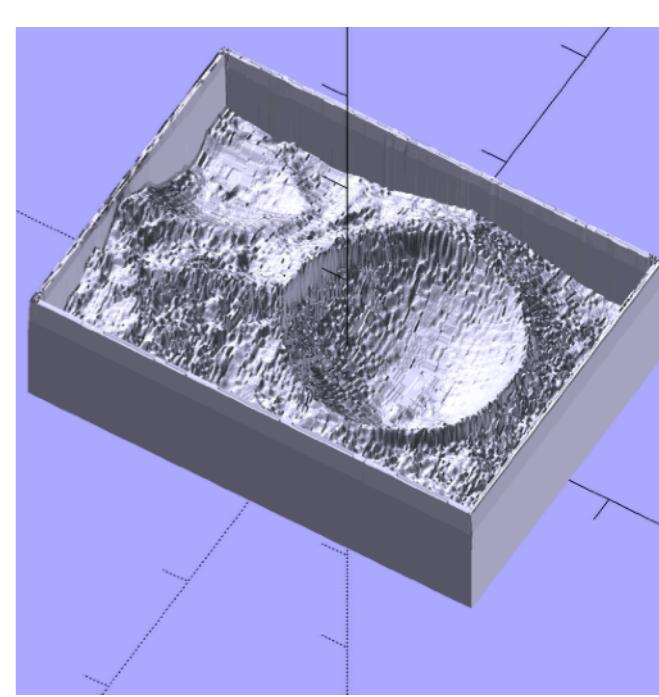
面白い機能2.5: 絵

- 白黒画像の濃淡を高さに読みかえて立体化してくれます。スタンプ製作などに便利!
- Heightmapと呼ばれるPNG画像をネットでさがして、そのまま立体化できます。

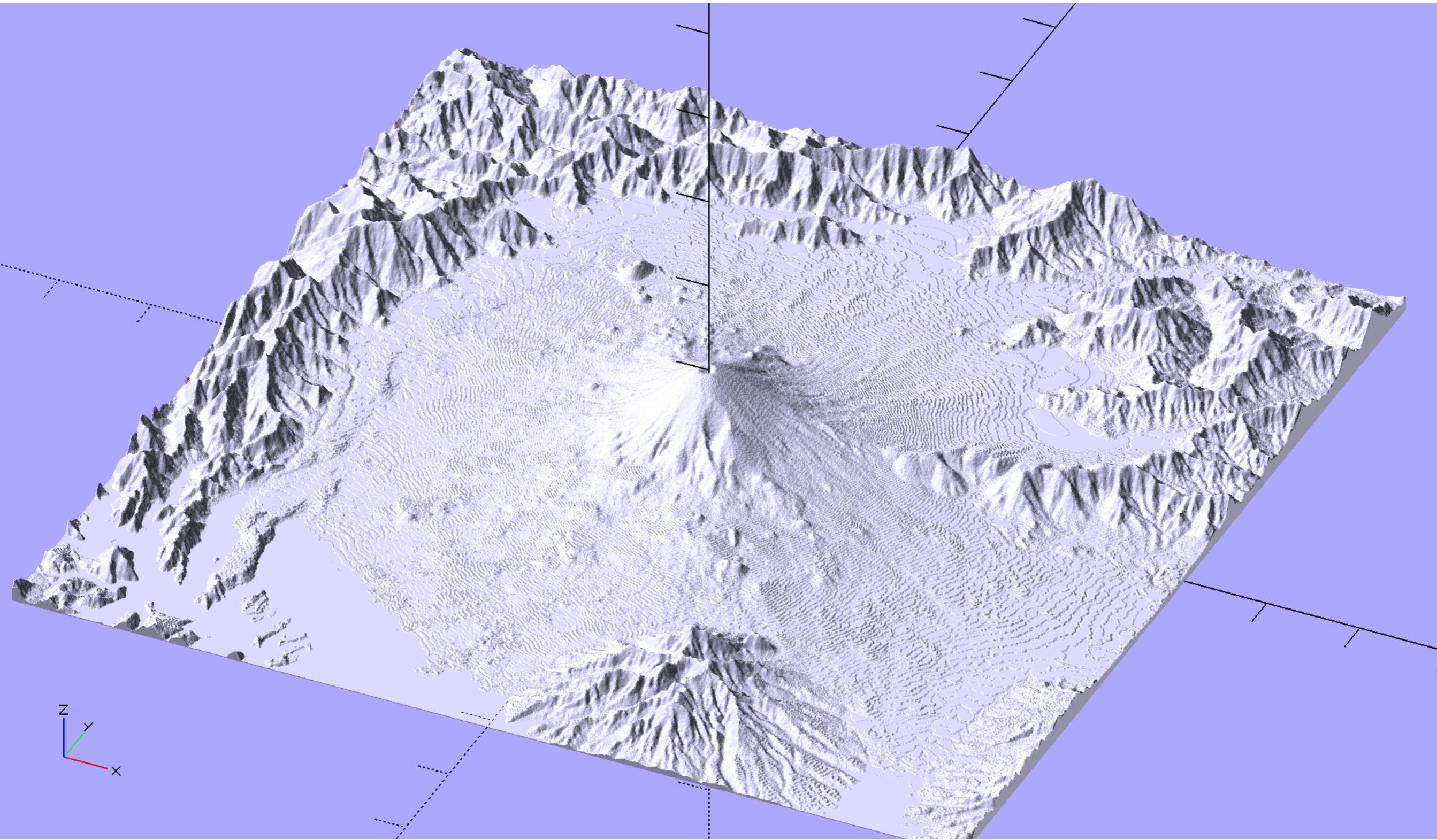
```
surface(file = "GoodLuck.png",
        center = true,
        invert = true);
```



月面クレーターの
heightmap



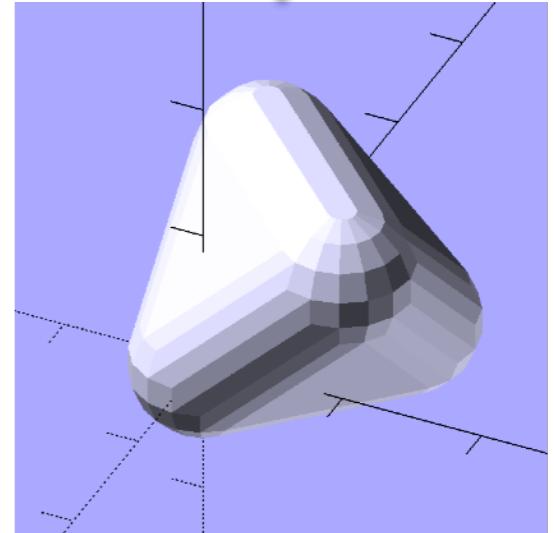
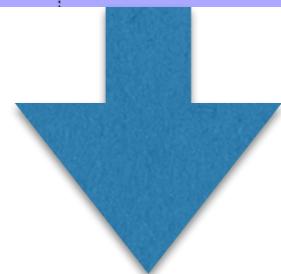
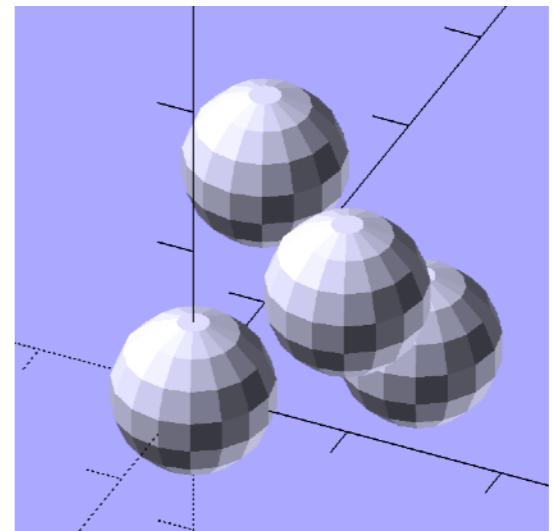
```
surface(file = "fuji.png", center = true, invert = false);
```



面白い機能3：凸包

- 凸包とは: 与えられた複数の物体をすべて含む、最小の凸多面体
- 角丸な立体を作るのにとても便利

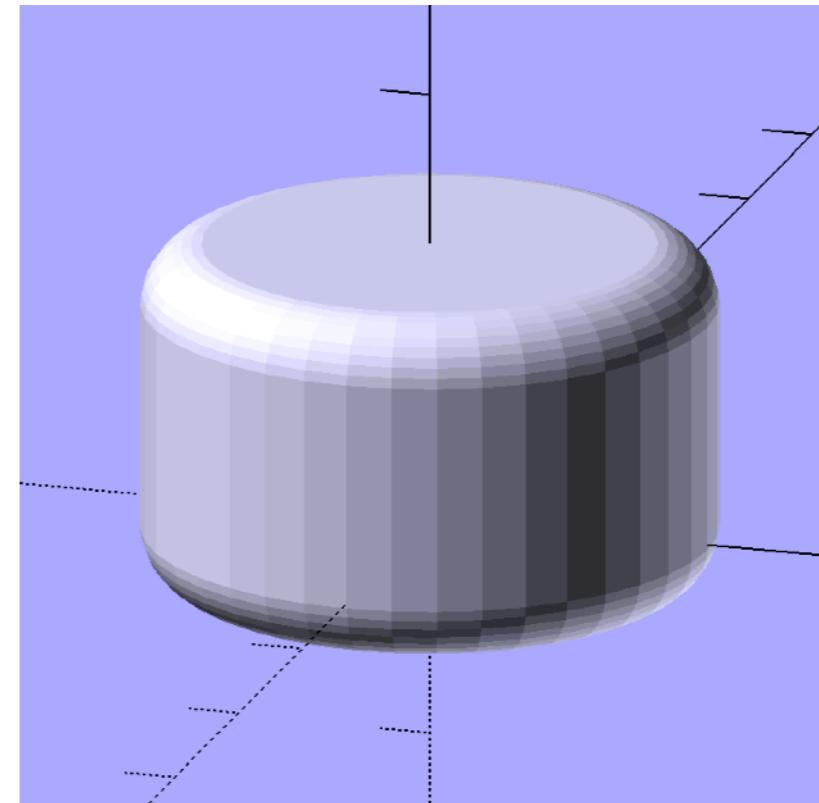
```
hull(){  
    translate([0, 0, 0]){sphere(r=5);}  
    translate([0,10,10]){sphere(r=5);}  
    translate([10,0,10]){sphere(r=5);}  
    translate([10,10,0]){sphere(r=5);}  
}
```



面白機能4：ミンコフスキーア

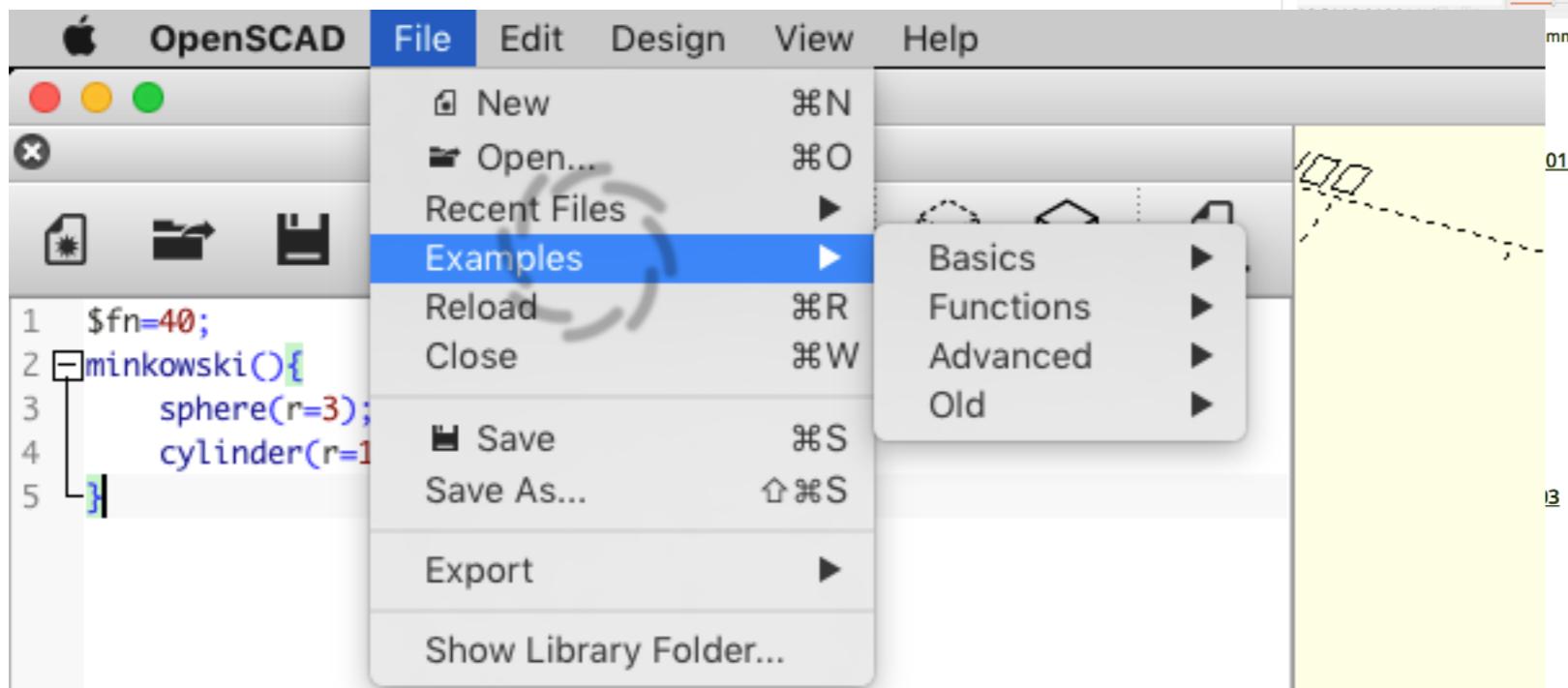
- 物体Aの形に沿って、物体Bを平行移動させたときにBが通る領域

```
$fn=40;  
minkowski(){  
    sphere(r=3);  
    cylinder(r=10,h=10);  
}
```



全部は説明しきれません!!

- OpenSCADをネット検索し、説明を見付けてみて下さい。
- <http://www.openscad.org>
- OpenSCADに附属のExampleを読んでみましょう



OpenSCAD
The Programmers Solid 3D CAD Modeler

Recent News

14 Jul 2016
[OpenSCAD Customizer](#)

OpenSCAD is a software for creating solid 3D CAD objects

It is free software and available for Linux/UNIX, MS Windows, Mac OS X and many others.

example009.scad

```

1 bodywidth = dxf_dim(file = "example009.dxf", name = "bodywidth");
2 fanwidth = dxf_dim(file = "example009.dxf", name = "fanwidth");
3 platewidth = dxf_dim(file = "example009.dxf", name = "platewidth");
4 fan_side_center = dxf_cross(file = "example009.dxf", layer = "fan_side_center");
5 fanrot = dxf_dim(file = "example009.dxf", name = "fanrot");
6
7 linear_extrude(height = bodywidth, center = true, convexity = 10);
8 import(file = "example009.dxf", layer = "body");
9
10 % for (z = [-bodywidth/2 + platewidth/2, -bodywidth/2 + platewidth/2];
11 %      z < bodywidth/2 + platewidth/2) {
12 %     translate([0, 0, z]);
13 %     linear_extrude(height = platewidth, center = true, convexity = 10);
14 %     import(file = "example009.dxf", layer = "plate");
15 %
16
17 intersection() {
18   linear_extrude(height = fanwidth, center = true, convexity = 10, twist = -fanrot);
19   import(file = "example009.dxf", layer = "fan_top");
20
21   // NB! We have to use the deprecated module here since the "fan_side"
22   // layer contains an open polyline, which is not yet supported
23   // by the import() module.
24   rotate_extrude(file = "example009.dxf", layer = "fan_side",
25                 origin = fan_side_center, convexity = 10);
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
    
```

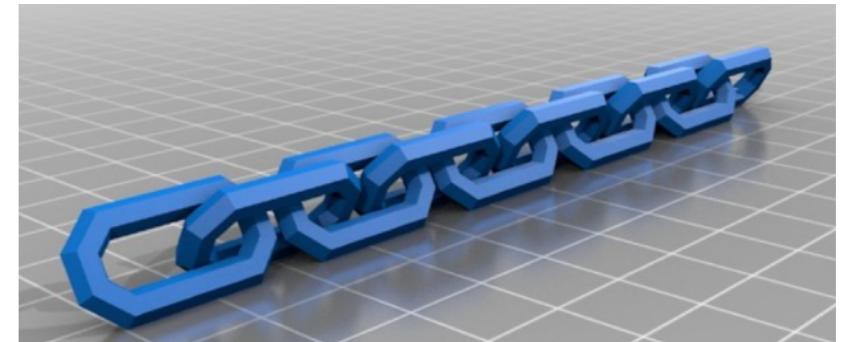
Viewport: translate = [0.61 -1.31 -2.07], rotate = [55.00 0.00 25.00], distance = 142.23

Download OpenSCAD
OpenSCAD 2015.03-3 Mac OS X

Other OSs and Versions

設計のヒント

- サポートができるだけ不要な形・向きを考えよう.
 - 例: テーブル, 鎖
- 2つのパツに分けると, シンプルに作れるかも.
- 素材の性質を知ろう



課題

- OpenSCADを使い、次のいずれかを自分でデザインして下さい。
 - ボールペンのボディ
 - チョコ用の型枠
 - スマホ用スタンド
 - 関節を持つ物体
 - 任意の物体（できれば $3\text{ cm} \times 3\text{ cm} \times 3\text{ cm}$ 以内で）

課題の提出

- 今日または今週中に、通常のレポートの代わりに,
scadファイルとstlファイルを提出して下さい。
(USBまたはメール)
宛先: vitroid@gmail.com 理学部本館B-133室
- 当日プリントできなかったものは、後日ひきとりに
来て下さい。 (プリントに2, 3日かかります)