Programming 3D models with OpenSCAD

@joewalnes

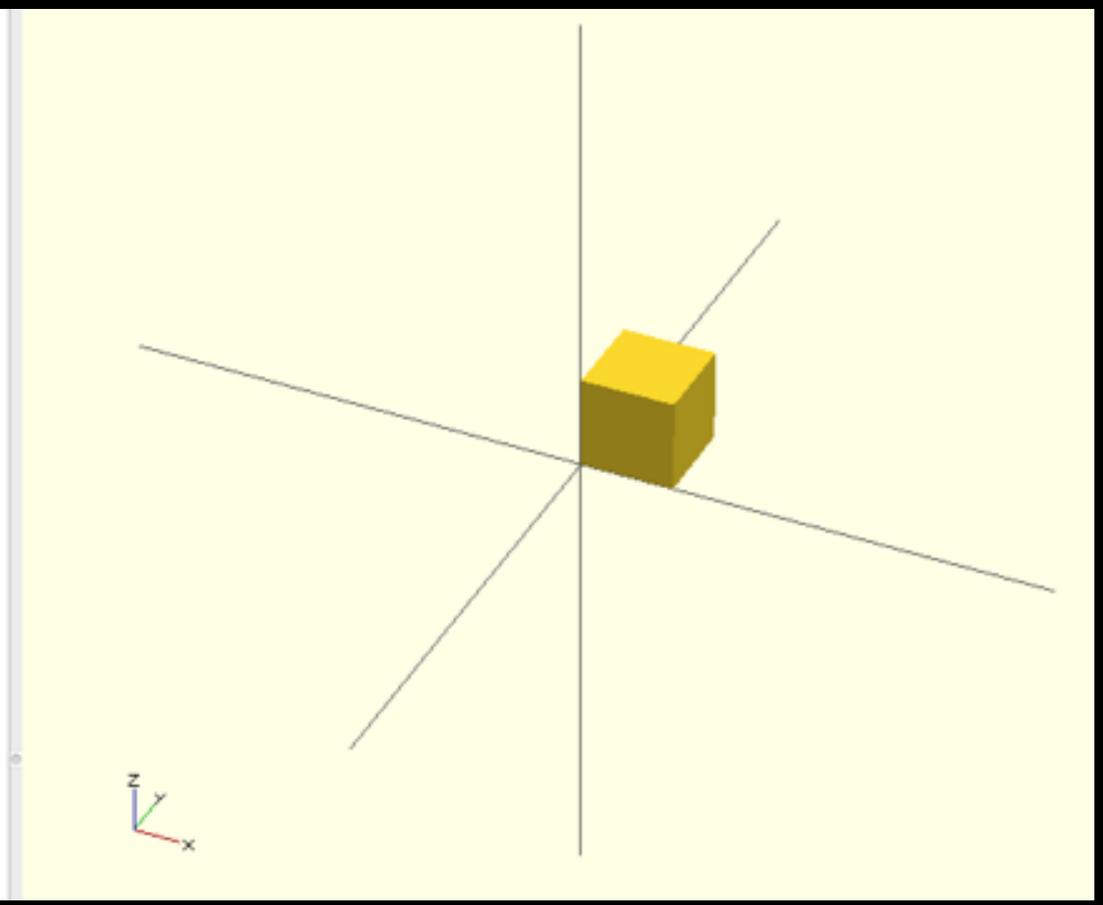
Just four concepts

1. Primitives

Start with something simple

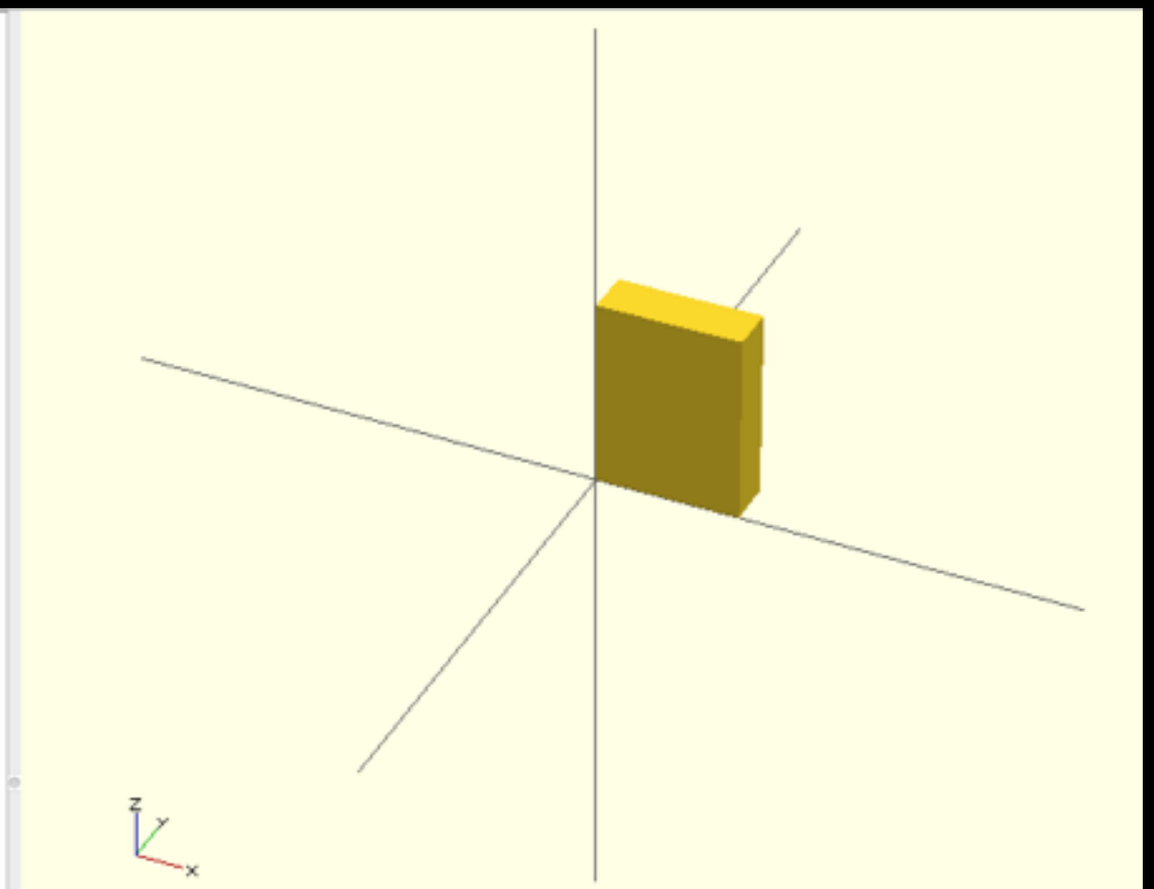
Cube

```
cube(10);
```



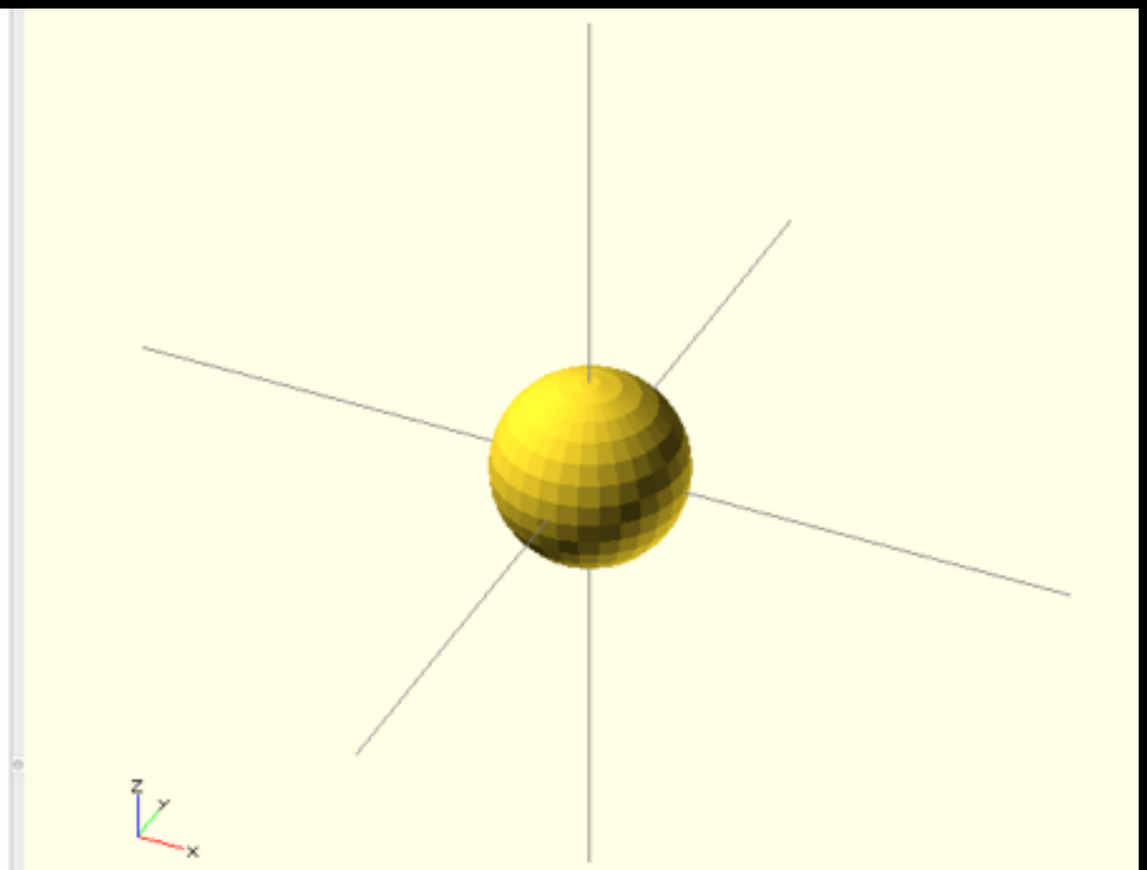
Irregular Cube

```
cube([15, 5, 20]);
```



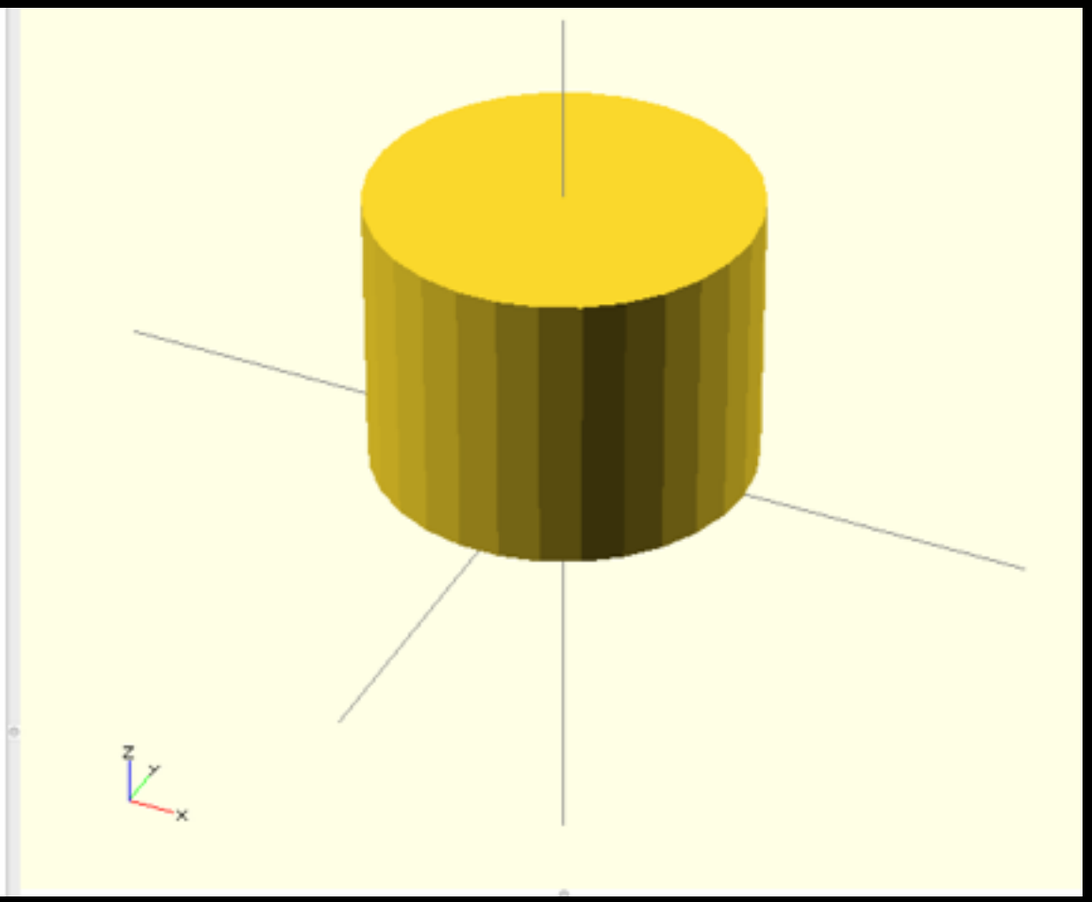
Sphere

```
sphere(d=20);
```



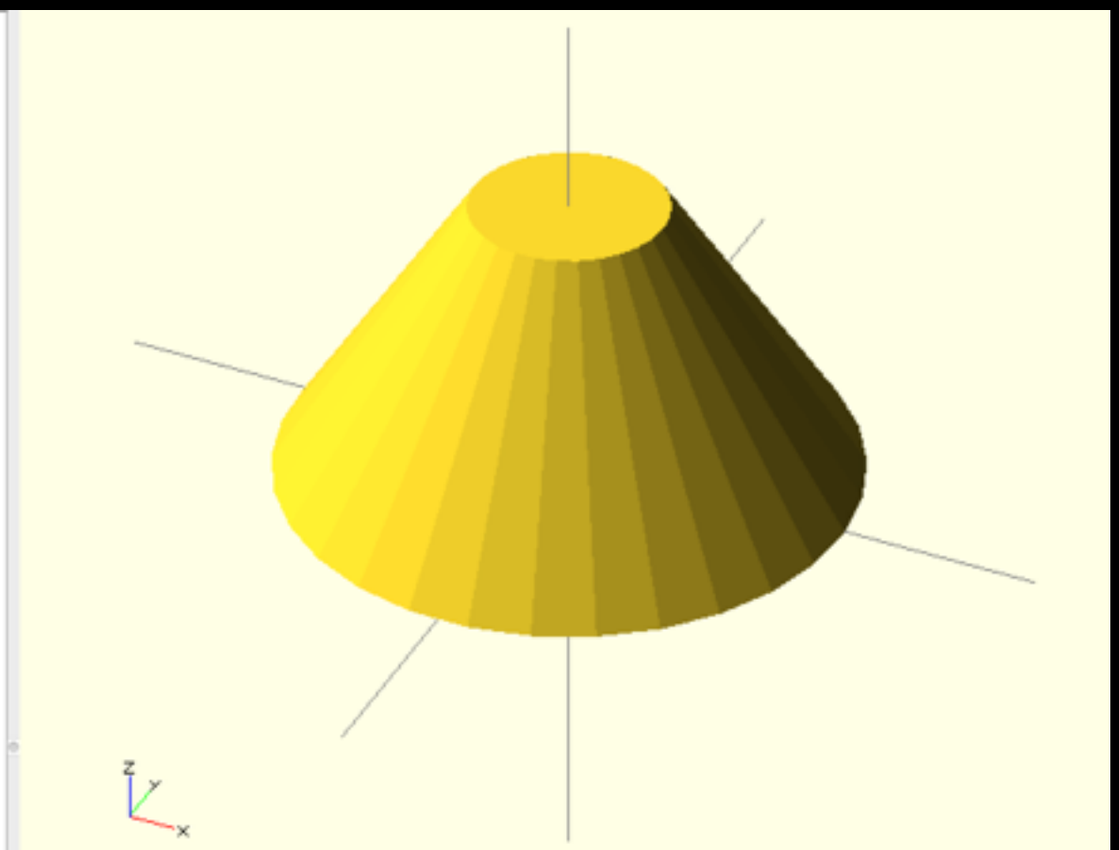
Cylinder

```
cylinder(h=30, r=20);
```



Weird Cylinder

```
cylinder(h=30,
         r1=30, r2=10);
```

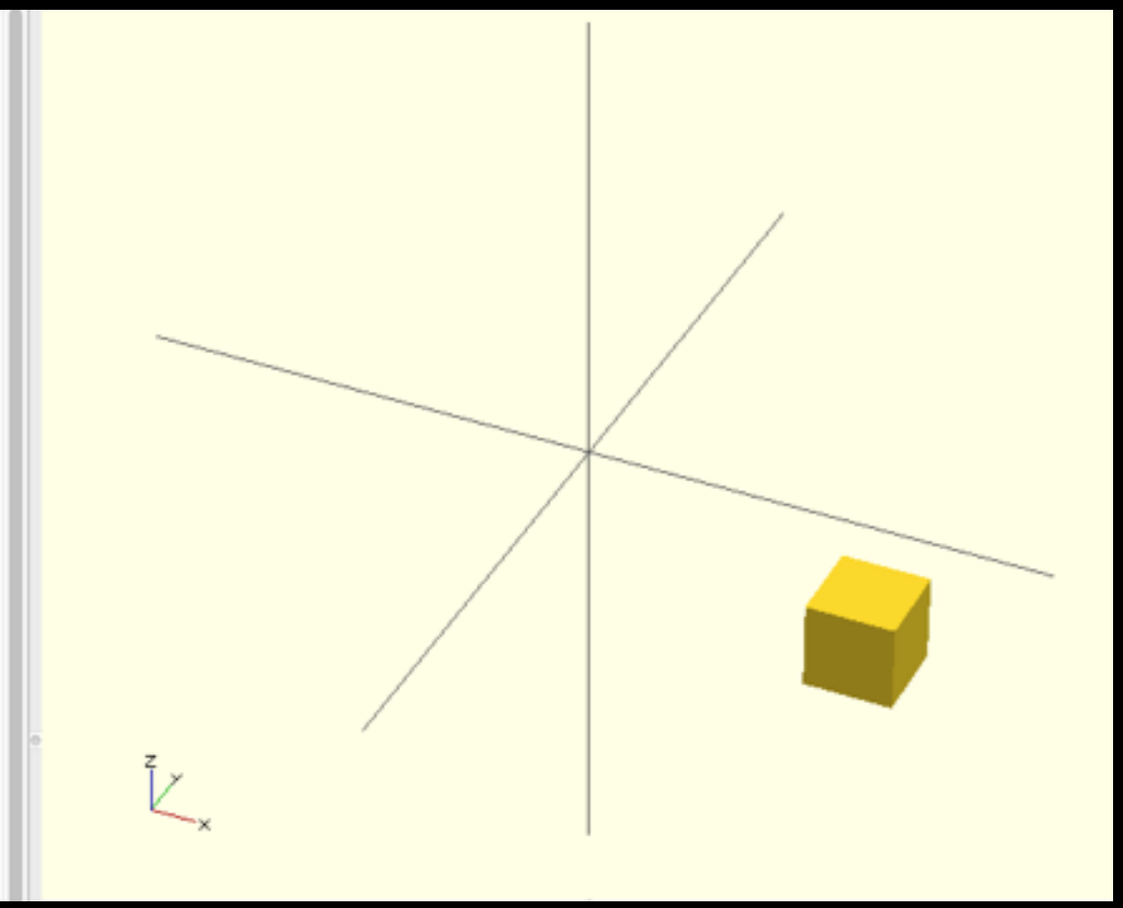


2. Transformations

Change it

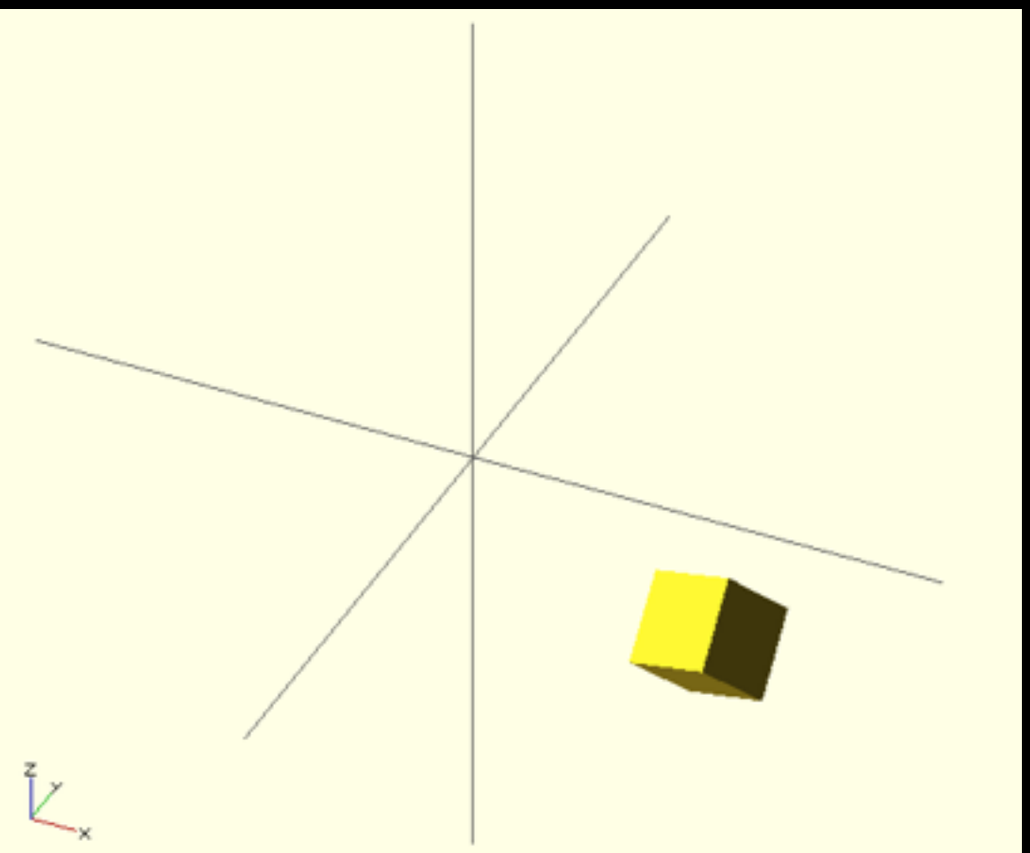
Translate

```
translate([20, 10, -30]) {
  cube(10);
}
```



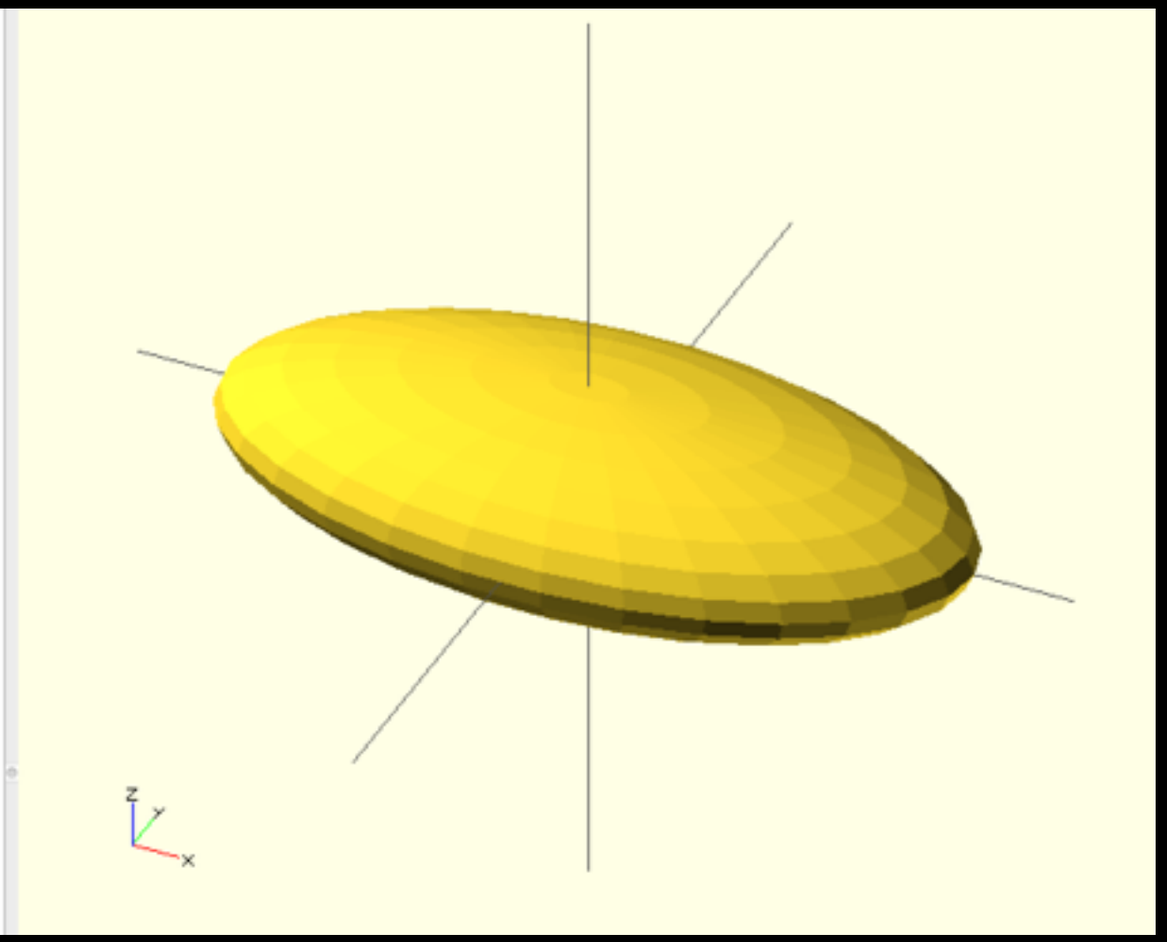
Rotate

```
translate([20, 10, -30]) {
   rotate([45, -20, -15.5]) {
      cube(10);
   }
}
```



Scale

```
scale([2.0, 1.0, -0.5]) {
   sphere(r=20);
}
```



3. Constructive Solid Geometry

Combining stuff

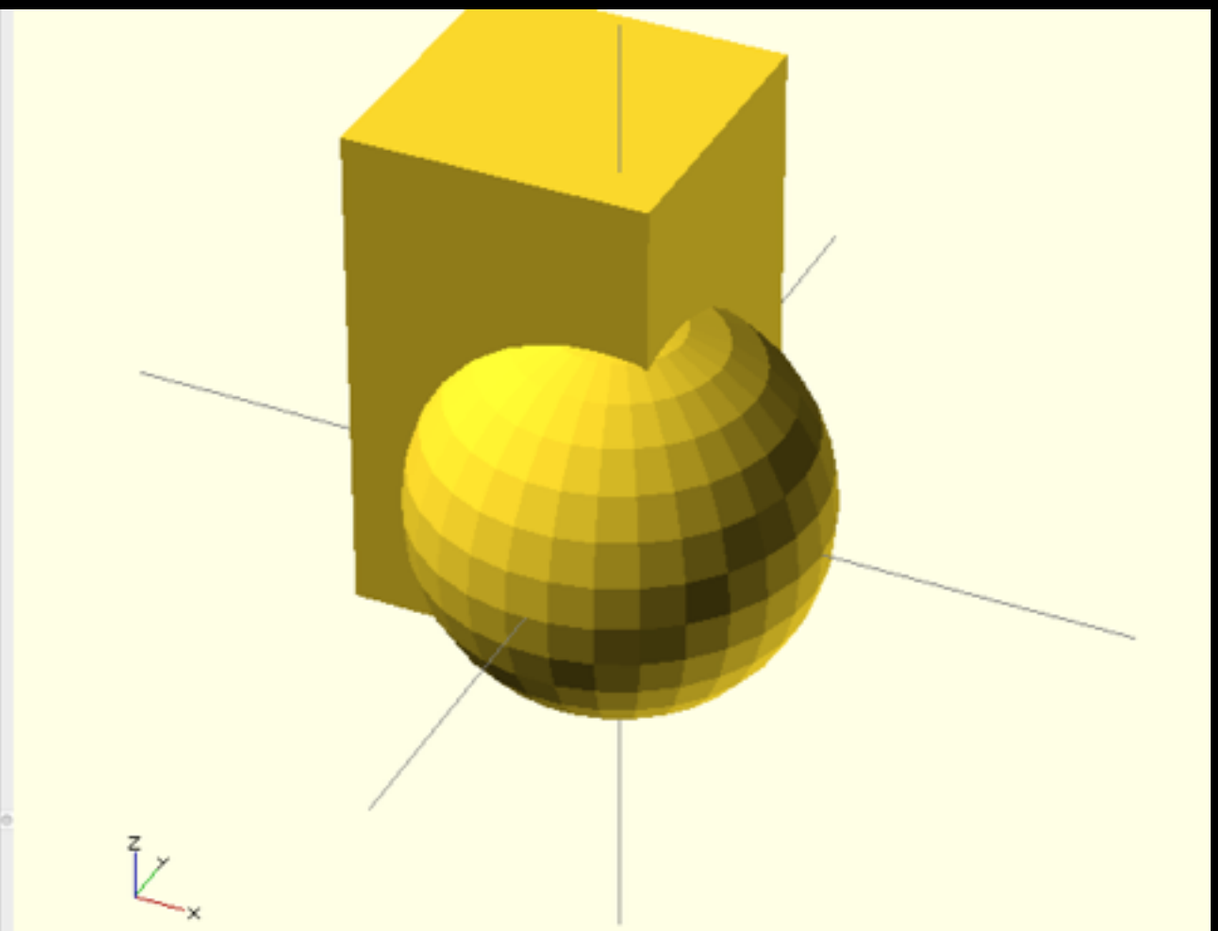
Union

```
union() {

  translate([-25, -5, -15]){
    cube([30, 30, 50]);
  }

  sphere(r=20);

}
```



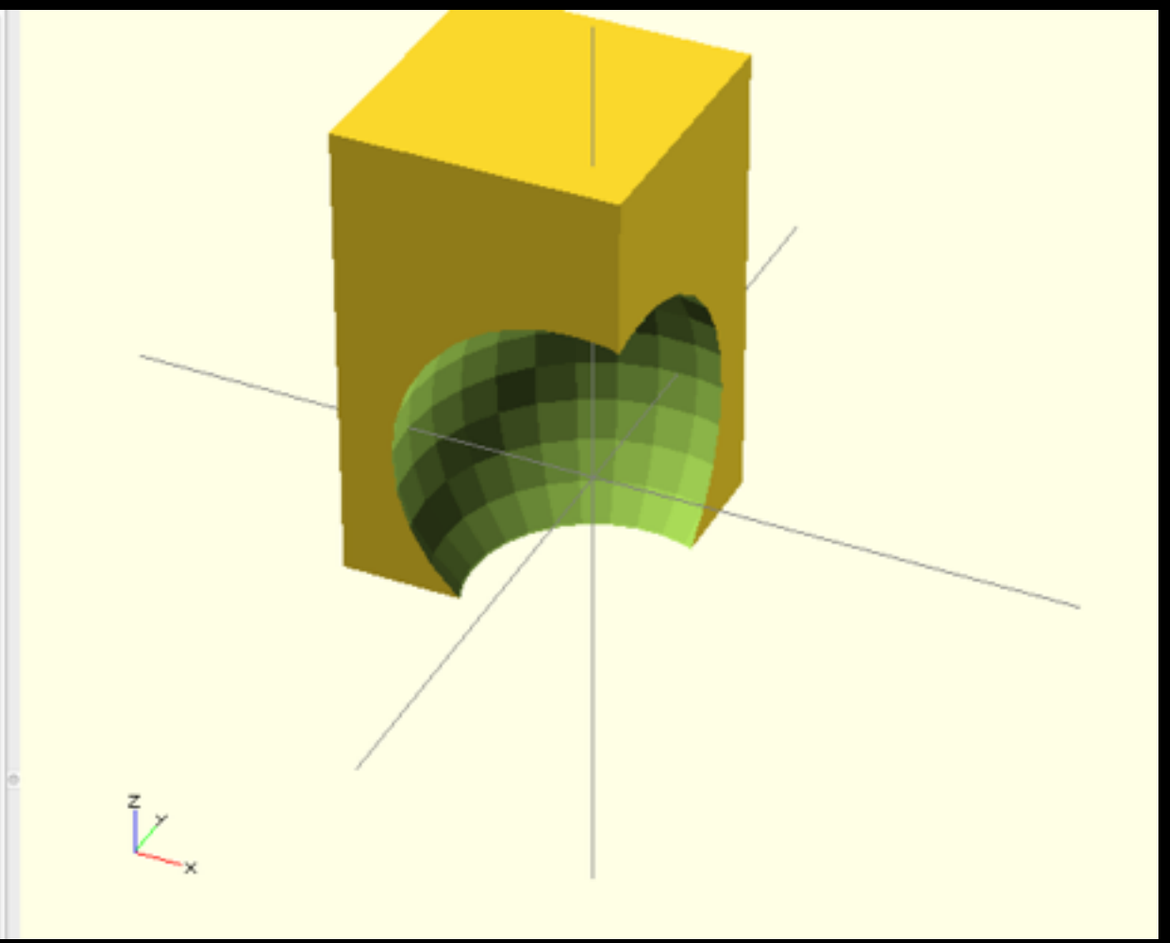
Difference

```
difference() {

  translate([-25, -5, -15]){
    cube([30, 30, 50]);
  }

  sphere(r=20);

}
```



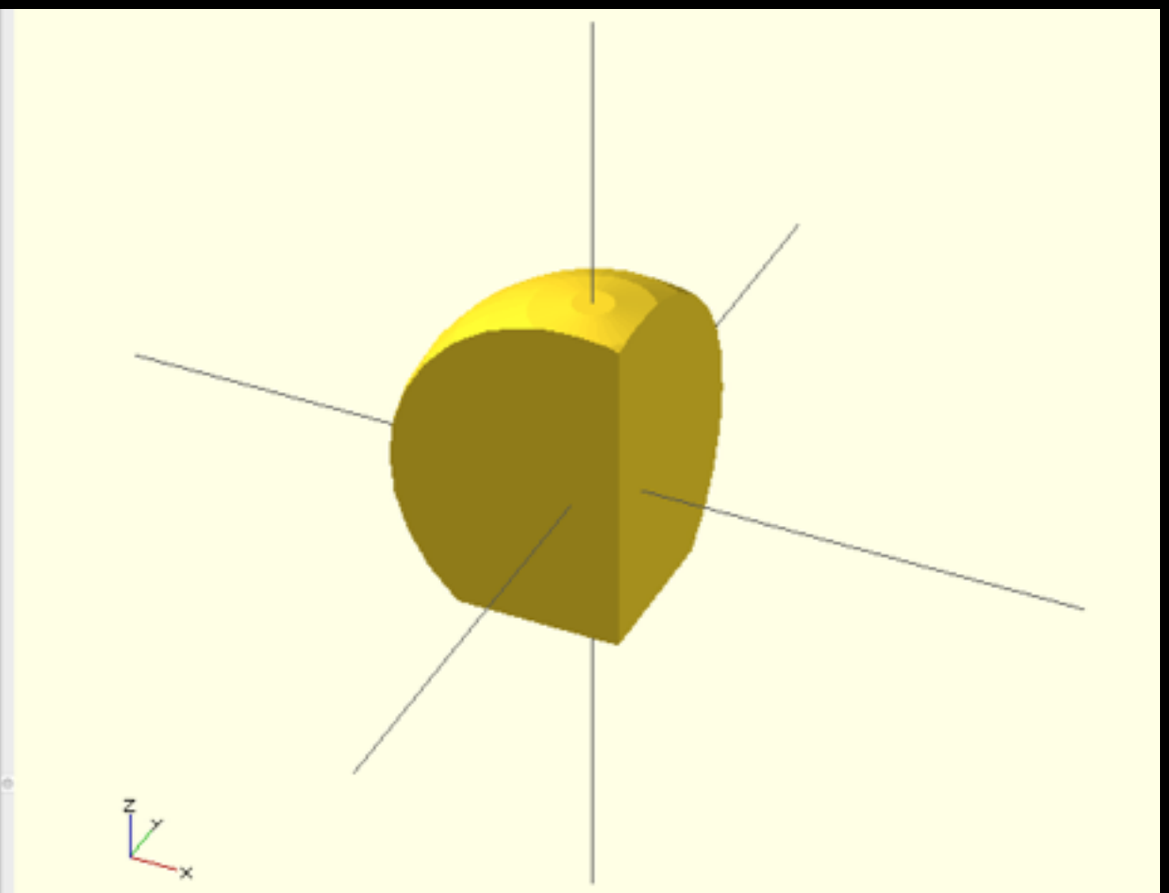
Intersection

```
intersection() {

  translate([-25, -5, -15]){
    cube([30, 30, 50]);
  }

  sphere(r=20);

}
```



4. Reuse

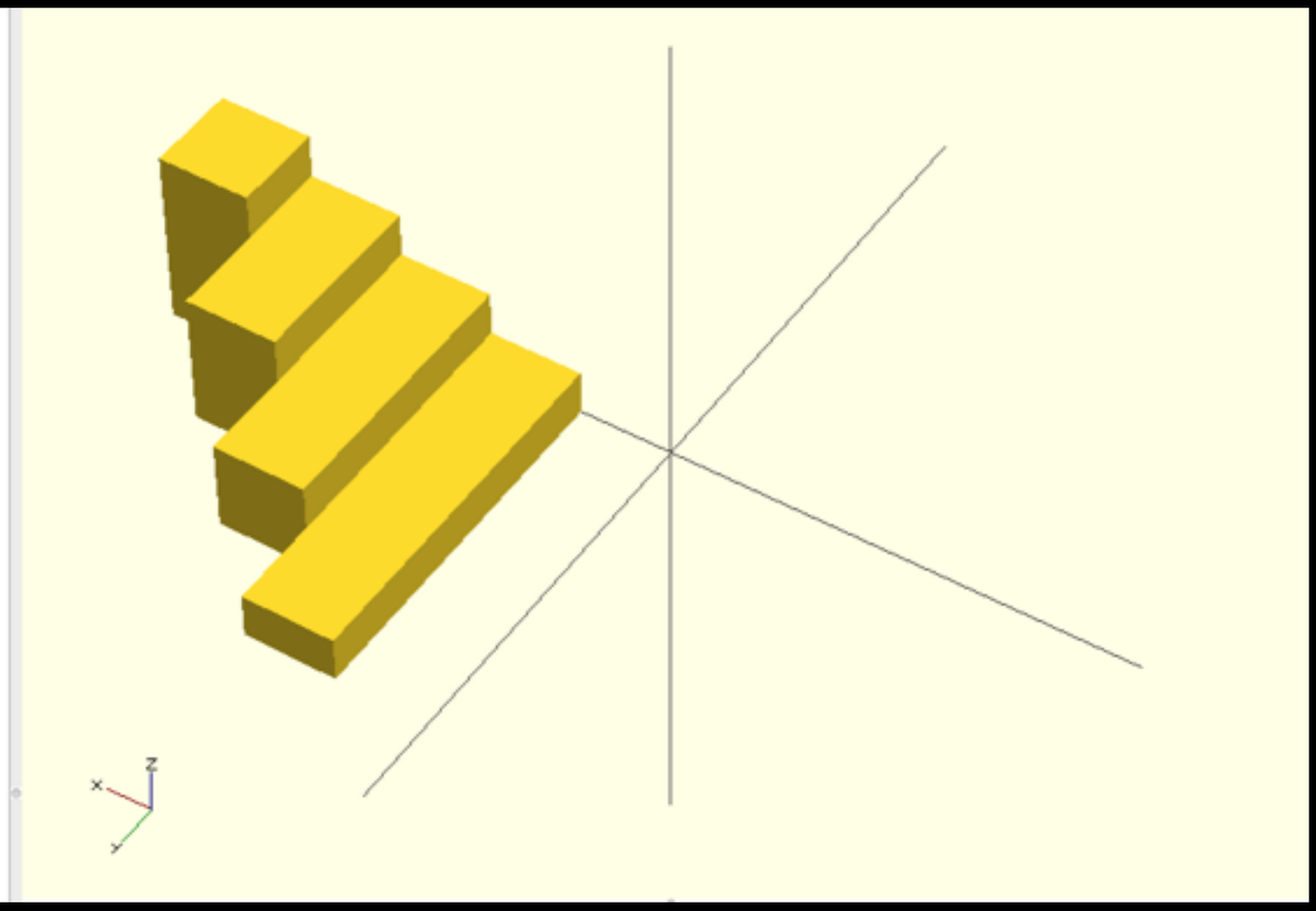
You know, it’s programming!

Loops, variables, etc

```
for (i = [0:5]) {

  translate(
    [i * 10, 0, 0]) {

    cube([
      10,
      50-i*10,
      5 * i]);
  }
}
```

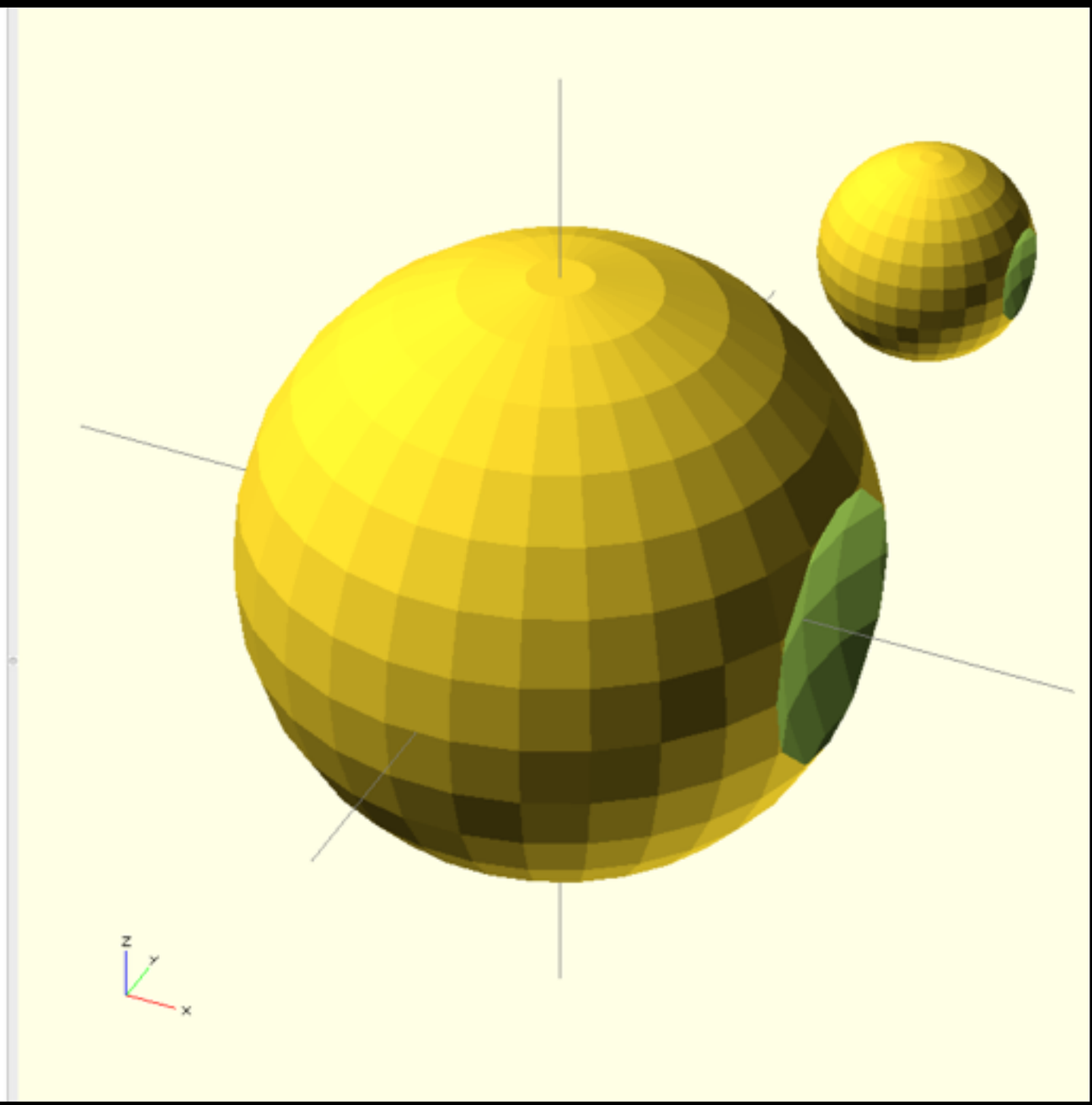


Modules

```
module deathstar(r) {
  difference() {
    sphere(r);
    translate(
      [r*1.8,0,0]){
        sphere(r);
    }
  }
}

deathstar(30);

translate([25, 25, 25]){
  deathstar(10);
}
```



Open Source Libraries

- Gears
- Screws
- Mounts
- Shapes
- Math functions
- Connectors
- .....

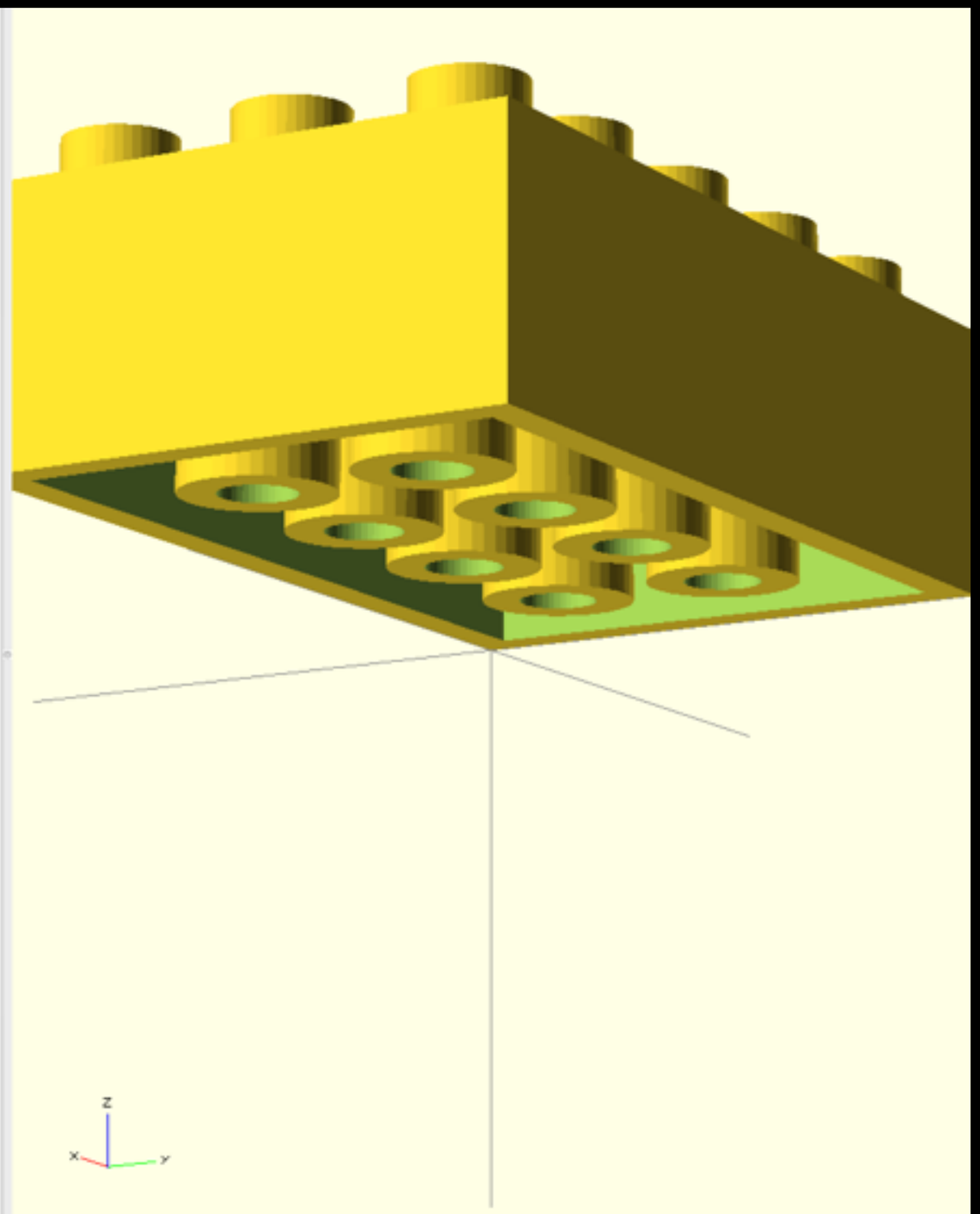
Put it all together…

```
// Make 5x3 brick. Change values for different sizes.
brick(5, 3);

module brick(units_wide, units_long) {
   body(units_wide, units_long);
   for (x=[0 : units_wide - 1], y=[0 : units_long - 1]) {
      stud(x, y);
      if (x > 0 && y > 0) {
         tube(x, y);
      }
   }
}

module body(units_wide, units_long) {
   difference() {
      cube([
            units_wide * length,
            units_long * length,
            height]);
      translate([wall_thickness, wall_thickness, 0]) {
         cube([
               units_wide * length - wall_thickness * 2,
               units_long * length - wall_thickness * 2,
               height - wall_thickness]);
      }
   }
}

module stud(unit_x, unit_y) {
   translate([
         (unit_x + 0.5) * length,
         (unit_y + 0.5) * length,
         height]) {
      cylinder(d=stud_diameter, h=stud_height);
   }
}
```



github.com/joewalnes/toybrick

Ok. Thanks. Go to openscad.org

-@joewalnes