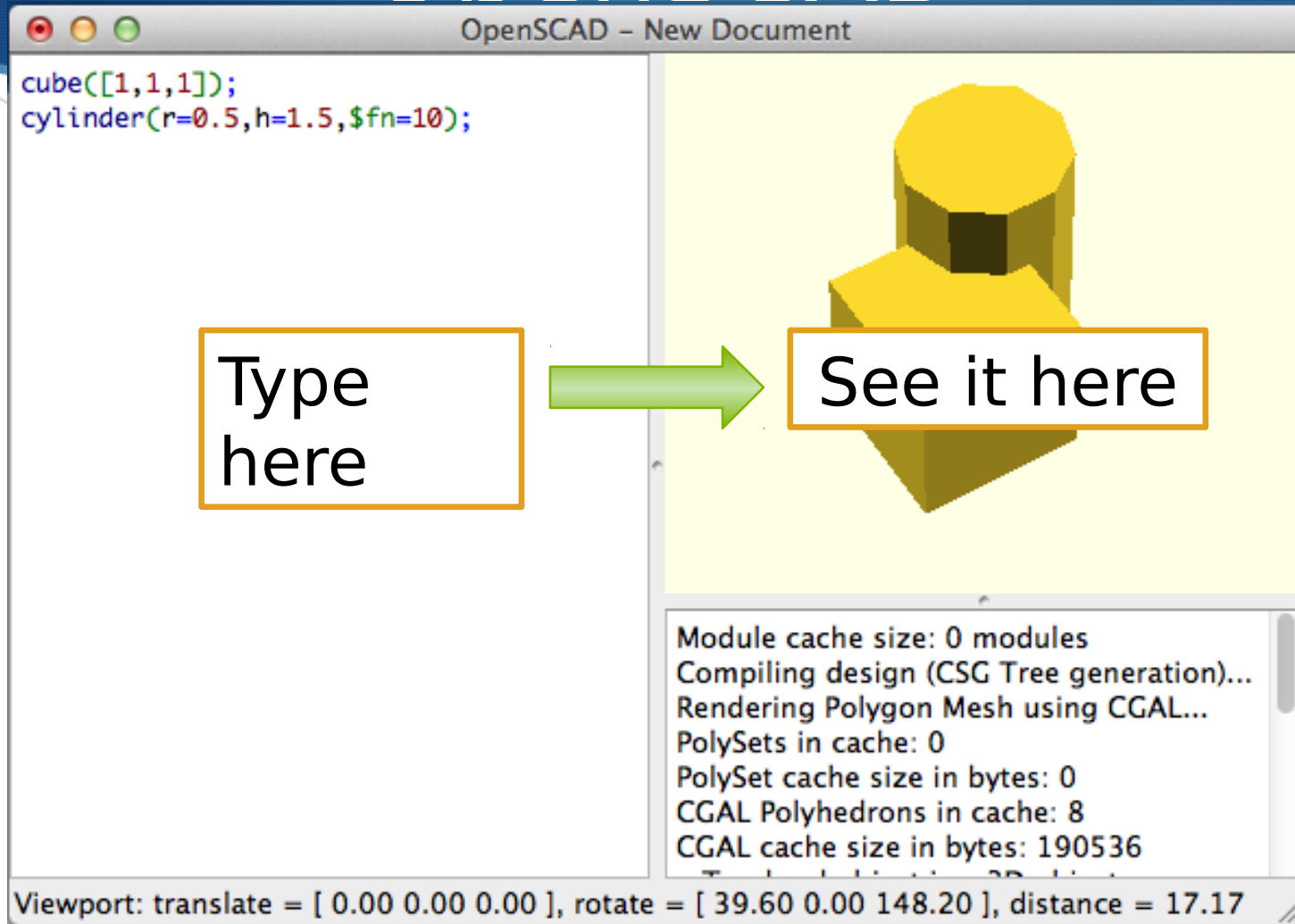


# Parametric Model Design for 3D Printing

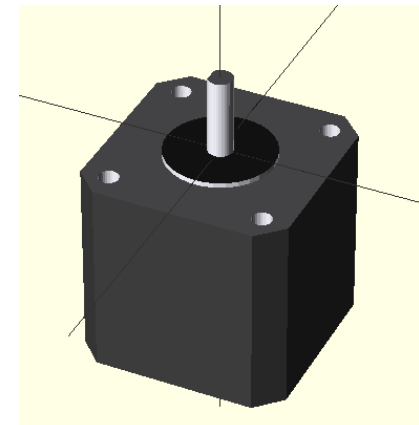
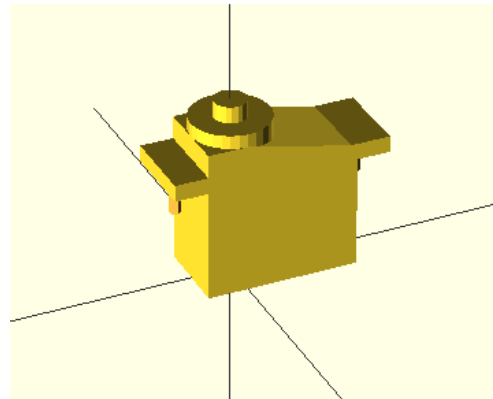
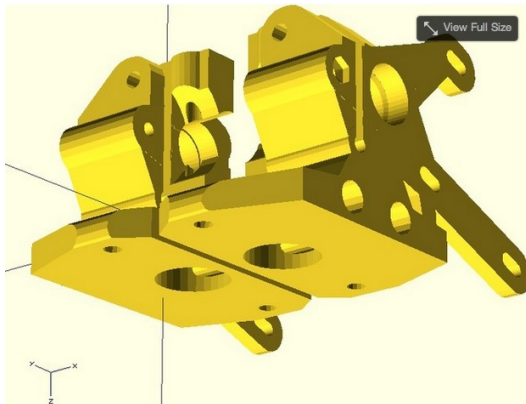
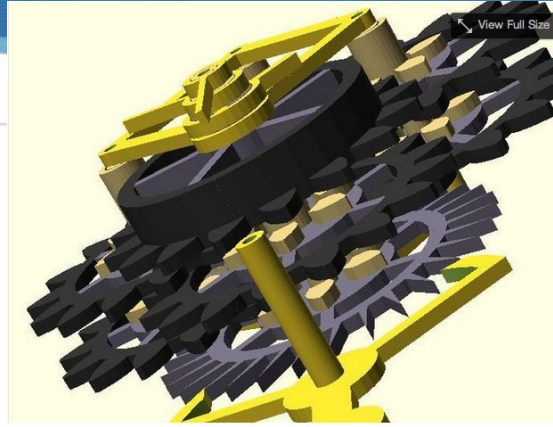
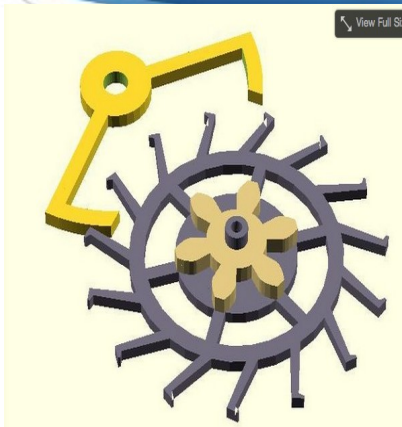
David S. Tyree [dtyree77@gmail.com](mailto:dtyree77@gmail.com)

# An introduction to OpenSCAD



# OpenSCAD is amazing!

(These are random things from thingiverse)

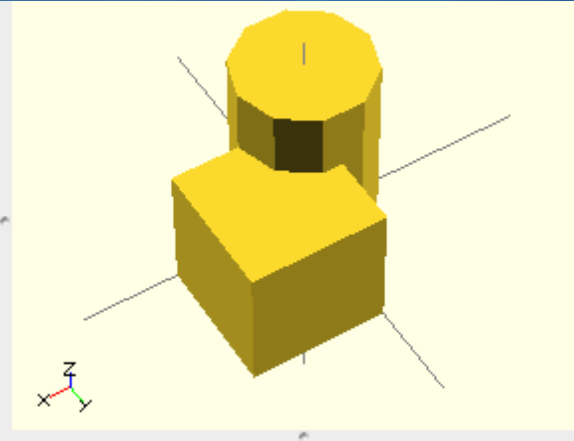


# Diving in!

- [www.openscad.org](http://www.openscad.org)
  - It's free!
  - It works on every major platform
  - Produces dimensionally accurate designs (by default all units are MM)
- To use it for 3D Printing
  - Type in your design
  - Render it
  - Save it to a .STL file
  - Use a Slicer to convert the .STL file to a .GCODE File
  - Print!

# Shapes

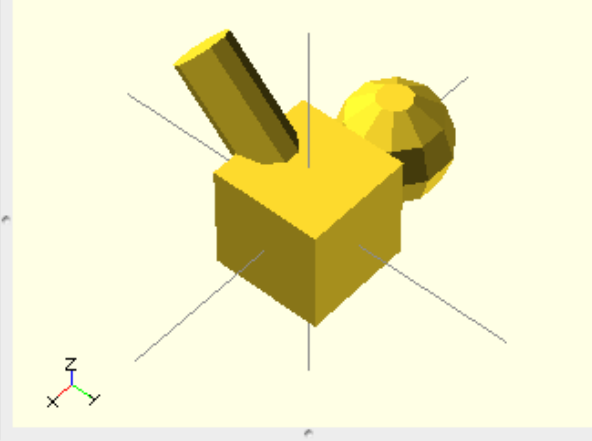
```
cube([1,1,1]);  
cylinder(r=0.5,h=1.5,$fn=10);
```



- You type in the shape's you want
  - `cube[x-dimension,y-dimension,z-dimension]);`
  - `cylinder(r=radius,h=height)`
  - `sphere(r=radius)`
- You can modify the shapes by adding:
  - `center=true` - center the shape
  - `$fn=fineness` - adjust the fineness of the generated shape
  - etc

# Modifying Shapes

```
$fn=10;  
cube([1,1,1],center=true);  
translate([-1,0,0]) sphere(r=0.5);  
rotate([45,0,0]) cylinder(r=0.25,h=1.5);
```



- You can modify a shape using these commands
  - `translate([moveByX,moveByY,moveByZ])`
  - `scale(newScale)` or `scale([scaleX,scaleY,scaleZ])`
  - `rotate([rotateX,rotateY,rotateZ])`
- Modify a shape by placing the command before it or by putting the shapes in braces '{ }'

# How to select what to modify,

- Modify a single shape

- `rotate([45,0,0]) translate([1,0,0]) cube([1,1,1]);`

- You can chain modifications

- The order matters! (move then rotate != rotate then move)

- First translate (move)

- Then rotate

- Modify a collection of stuff

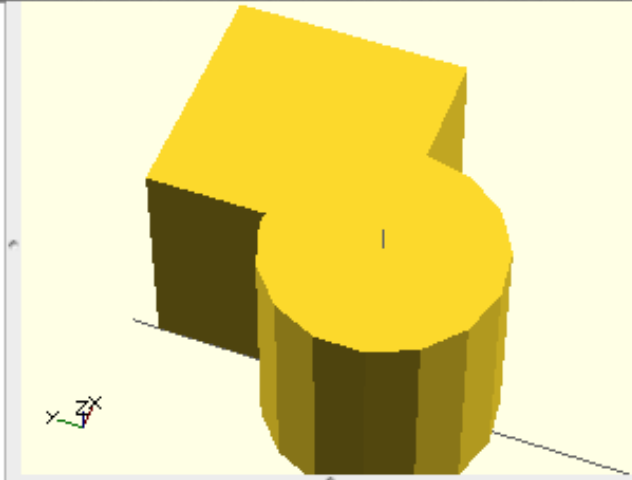
- `rotate([45,0,0]) translate([1,0,0]) {`  
    `cube(1,1,1);`  
    `scale(1.5) sphere(r=2);`  
□ `}`

- Operations are applied in reverse order

- `sphere(r=2)` is first scaled, then translated and finally rotated

# Union

```
$fn=15;  
union(){  
  cube([1,1,1]);  
  cylinder(r=0.5,h=1);  
}
```

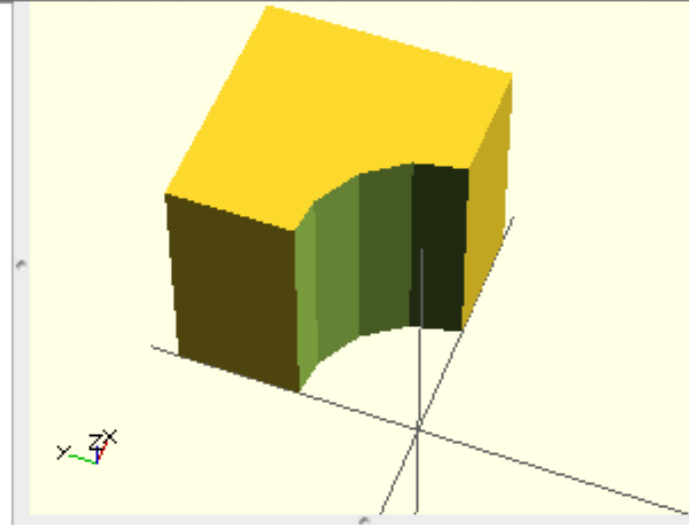


- `union(){ }`
- Produce a shape by combining all the shapes



# Difference

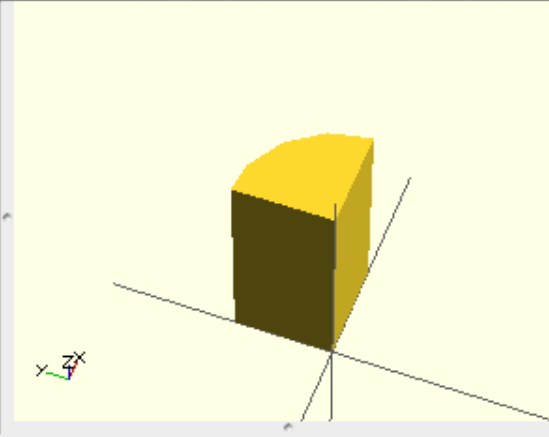
```
$fn=15;  
difference(){  
  cube([1,1,1]);  
  cylinder(r=0.5,h=1);  
}
```



- You can perform operations on shapes to create new shapes
  - `difference(){ shapeA(); shapeB(); shapeC(); }`
  - Subtract shapes from each other
  - `shapeA-(shapeB+shapeC)`

# Intersection

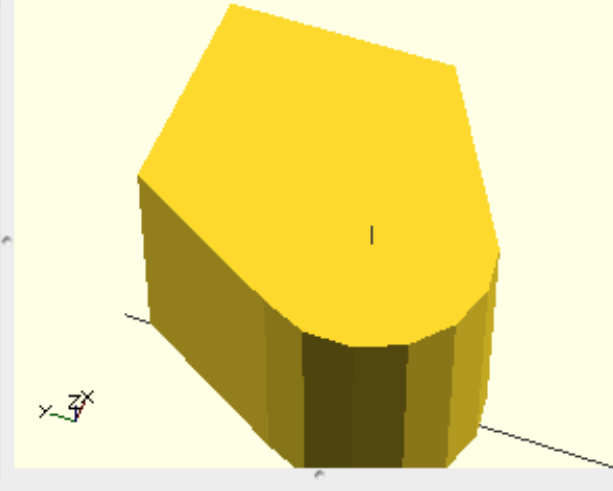
```
$fn=15;  
intersection(){  
  cube([1,1,1]);  
  cylinder(r=0.5,h=1);  
}
```



- `intersection(){ }`
- Product a new shape from the intersection of shapes

# Hull

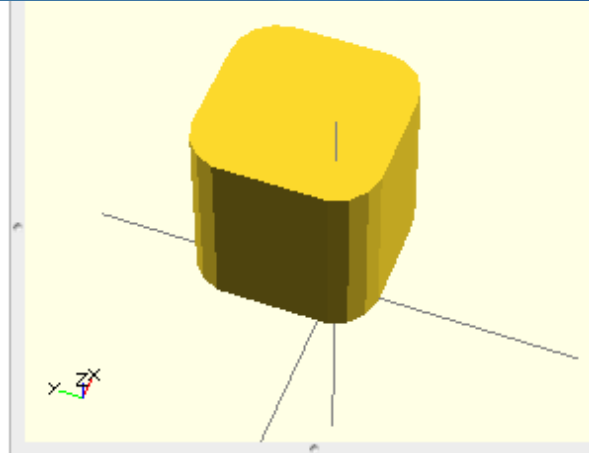
```
$fn=15;  
hull(){  
  cube([1,1,1]);  
  cylinder(r=0.5,h=1);  
}
```



- `hull(){ }`
- Produce a shape by combining the profiles of two shapes

# Minkowski

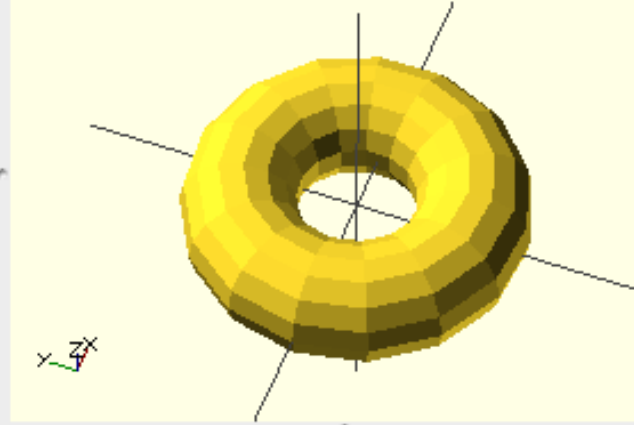
```
$fn=15;  
minkowski() {  
  cube([1,1,1]);  
  cylinder(r=0.5,h=1);  
}
```



- `minkowski() { }`
- Produce a shape by tracing one shape around another
  - This will trace the cylinder around the cube

# Rotate & Extrude

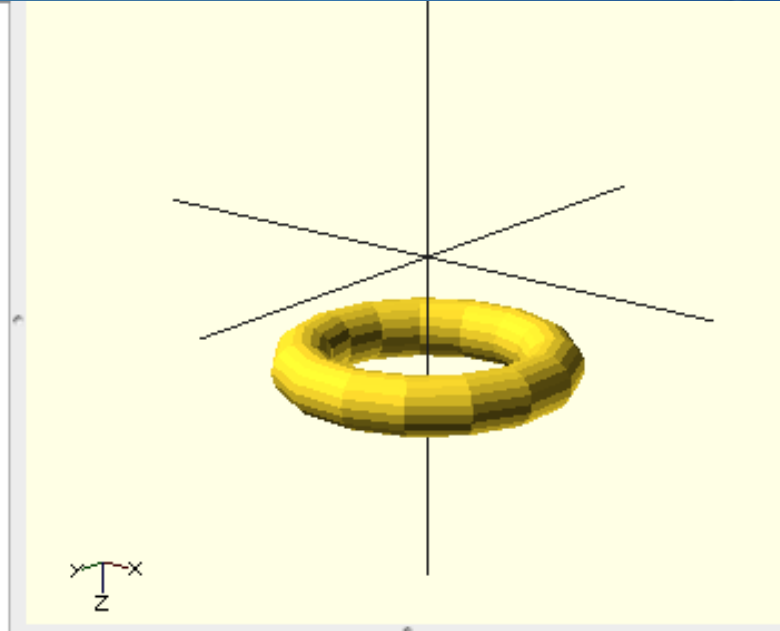
```
$fn=15;  
rotate_extrude(convexity = 10)  
translate([2, 0, 0])  
circle(r = 1);
```



- `rotate_extrude(){ }`
- Produce a shape rotate it and extruding it

# Modules

```
$fn=15;  
  
module groove()  
{  
    translate([0,0,15]){  
        rotate_extrude(convexity = 10)  
        translate([17, 0, 0 ])  
        circle(r = 4,$fn=20);  
    }  
}  
  
groove();
```



- Modules let you combine a bunch of shapes and operations into a single thing so you can re-use.
- They also let OpenSCAD cache a shape

# Composite shapes!

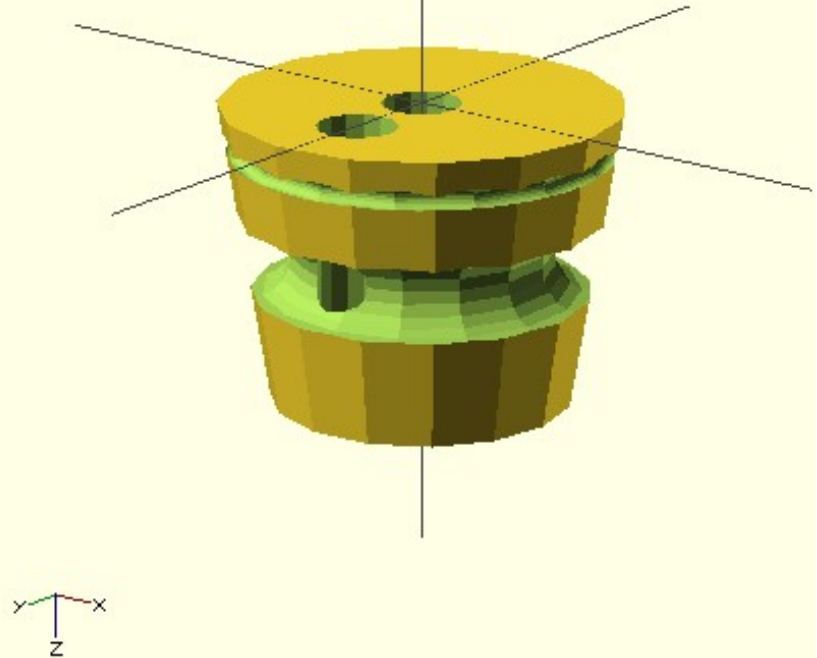
```
$fn=15;

module plug(){
  difference(){
    cylinder(r1=20,r2=15,h=30);
    union(){
      groove();
      oring();
      translate([0,10,-1]) cylinder(r=4,h=22);
      translate([0,0,-1]) cylinder(r=4,h=35);
    }
  }
}

module groove(){
  translate([0,0,15]){
    rotate_extrude(convexity = 10)
    translate([17, 0, 0 ])
    circle(r = 4,$fn=20);
  }
}

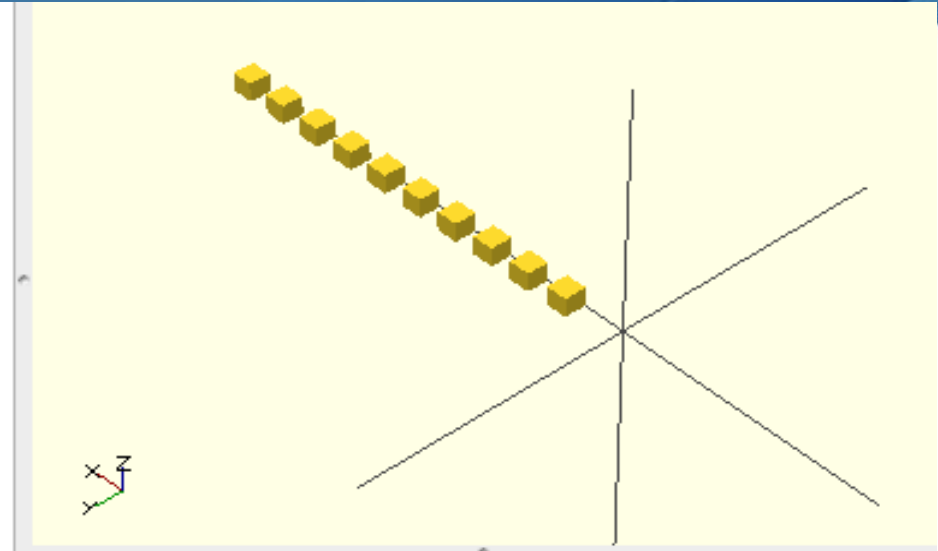
module oring(){
  translate([0,0,4]){
    rotate_extrude(convexity = 10)
    translate([19, 0, 0 ])
    circle(r = 1,$fn=20);
  }
}

plug();
```



# Loop & Iterate

```
$fn=15;  
for(i=[1:10]){  
    translate([i*2,0,0]) cube([1,1,1]);  
}
```



- OpenSCAD has variables
- `i=5; //set i to 5`
- You can loop or iterate with openscad
- `for(variable=[start:increment:end]){`
  -
- `}`



# 3D IO

The power is in combinations

## □ Inputs

- 3D Model (.OBJ, .STL)

- Thingiverse, Sculpttris, Wings3D, 123D Catch, etc

- 2D Models / Drawings (.DXF, .SVG, PD)

- Inkscape, Autocad, QCAD, Desktop Scanner, etc

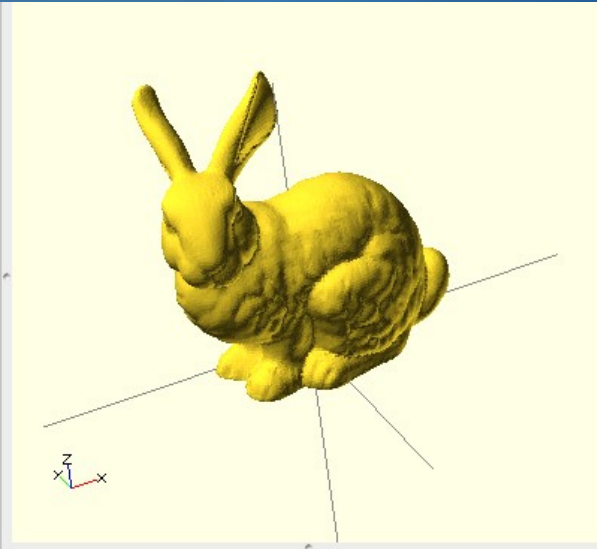
- OpenSCAD can import and use these files

- `import("kitten.obj");`

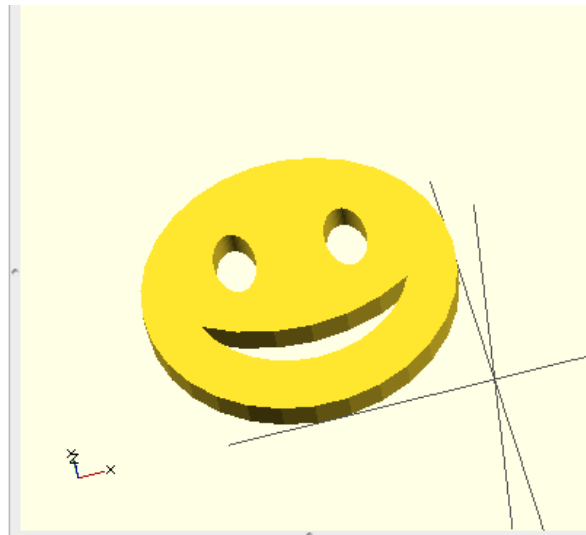
- You can then use them like any other shape!

# Importing things!

```
import("/Users/dtyree/Downloads/bunny-flatfoot.stl");
```



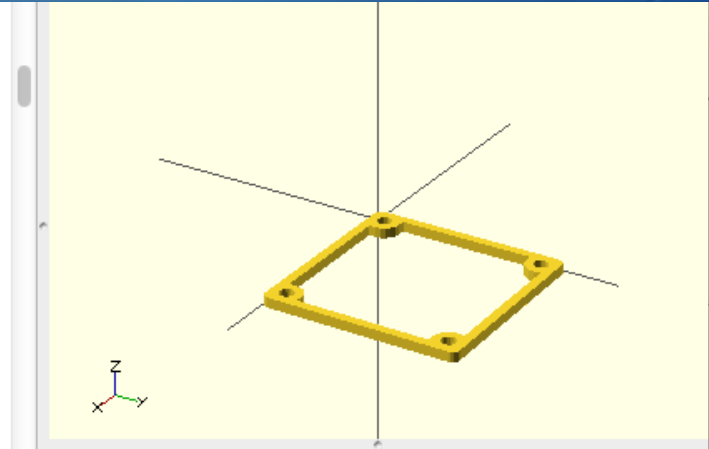
```
linear_extrude(height=5){  
  import("/Users/dtyree/Downloads/smiley.dxf");  
}
```



# Using libraries

```
//uncomment this for example
fan_mount(size=60,thick=3);

module fan_mount(size=40,thick = 4)
{
  if(size == 25)
  {
    _fan_mount(
      fan_size = 25,
      fan_mounting_pitch = 20,
      fan_m_hole_dia = 3,
      holder_thickness = thick
    );
  }
  if(size == 30)
  {
    _fan_mount(
```

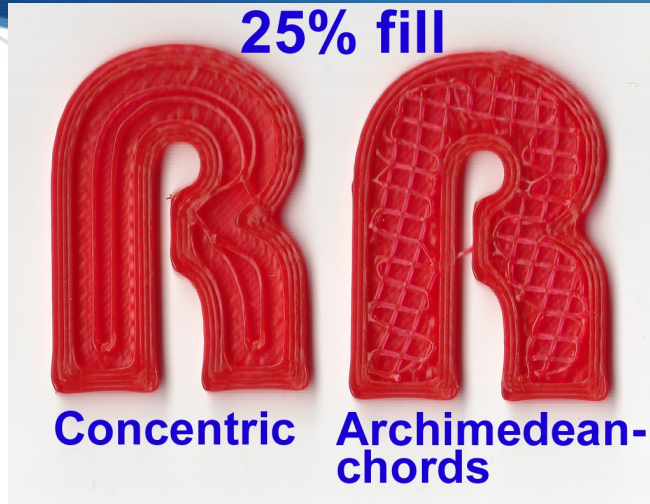


- OpenSCAD has a large number of libraries
  - MCAD ( <https://github.com/elmom/MCAD> )
  - Screws, gears, servos, steppers, motors, bolts, etc
  - Thingiverse
- Download the library and then import it
  - Usually the library will have instructions inside of how to use it

# Designing stuff!

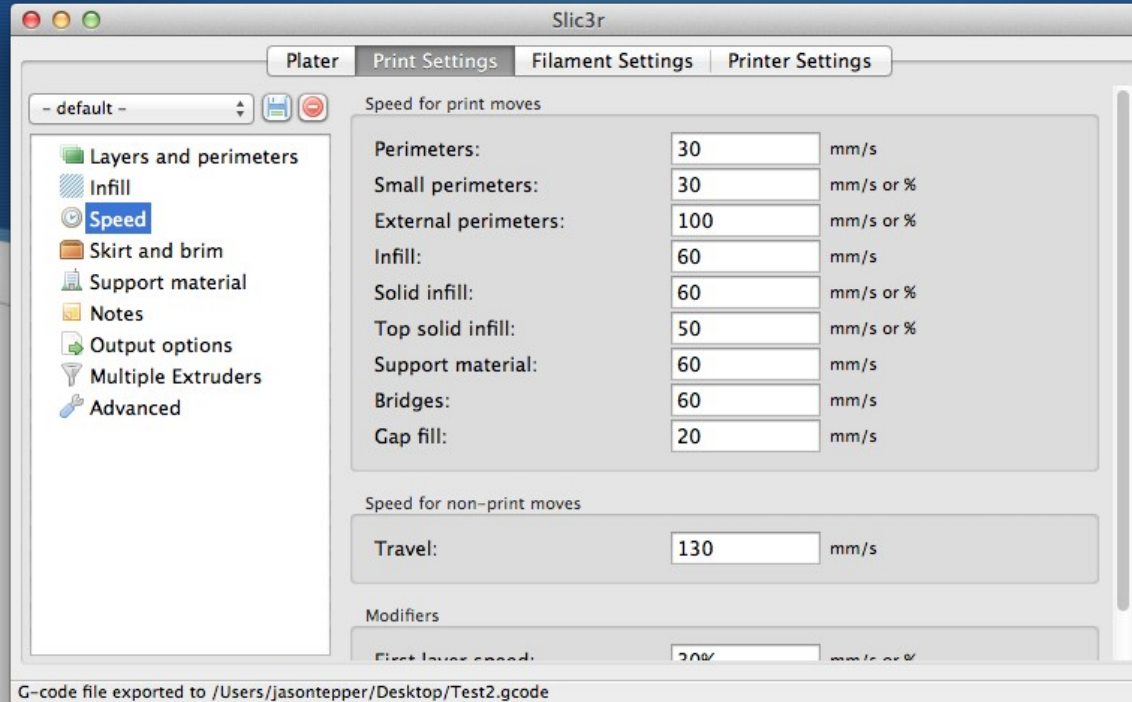
- What requirements?
- What tools / inputs do I have ?
  - Capture from real life
  - Desktop Scanner / 123D Catch
  - Model it
  - Sculptris, Thingiverse, Sketchup
- Can I print it?
  - Does it have big overhangs?
  - How dense does it need to be?
  - How accurately must it be printed to work (0.3mm is what I consider the minimum feature size to be useful)

# Slicing



- After creating the thing we choose some print settings

# Slicing



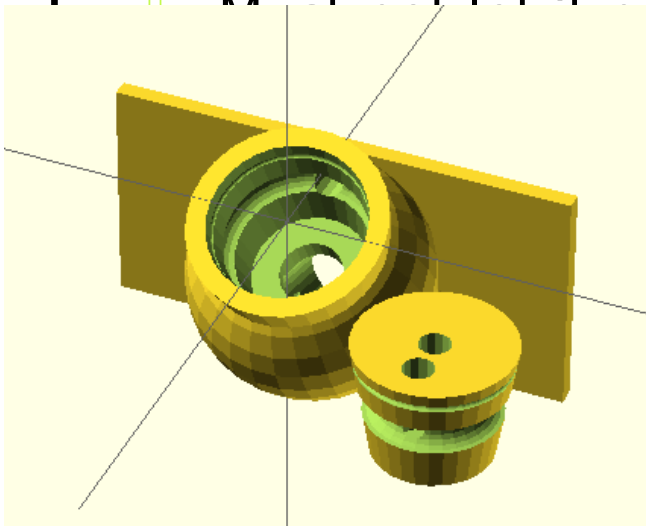
- After designing the model it is time to print it! (I use Slic3r)
  - Choose a density and fill pattern
  - Choose number of top / bottom layers
  - Choose number of perimeter
- Print the generated .GCODE file (I use Printron)



# Example: Quick disconnect for cool suit for racing

(Drawing it out on paper is a must for this one)

- A hose connector for quickly disconnecting hoses for a cool suit, which can be mounted on the side of a cooler
- Must disconnect quickly and easily
- Must plug back in easily



# Example: A custom JIG for drilling out a snapped bolt on a motor

- ❑ You want to build a JIG to drill out a snapped exhaust bolt.
- ❑ Use a desktop scanner to scan a new gasket for the exhaust
- ❑ Convert to STL file
- ❑ Extrude





# Example: A Fallout PitBoy

- You want to build a Fallout Pitboy
- You can use 123D catch to acquire a model of your wrist
- You can then use a 3D model of the PitBoy and subtract the model of your wrist from it.

