

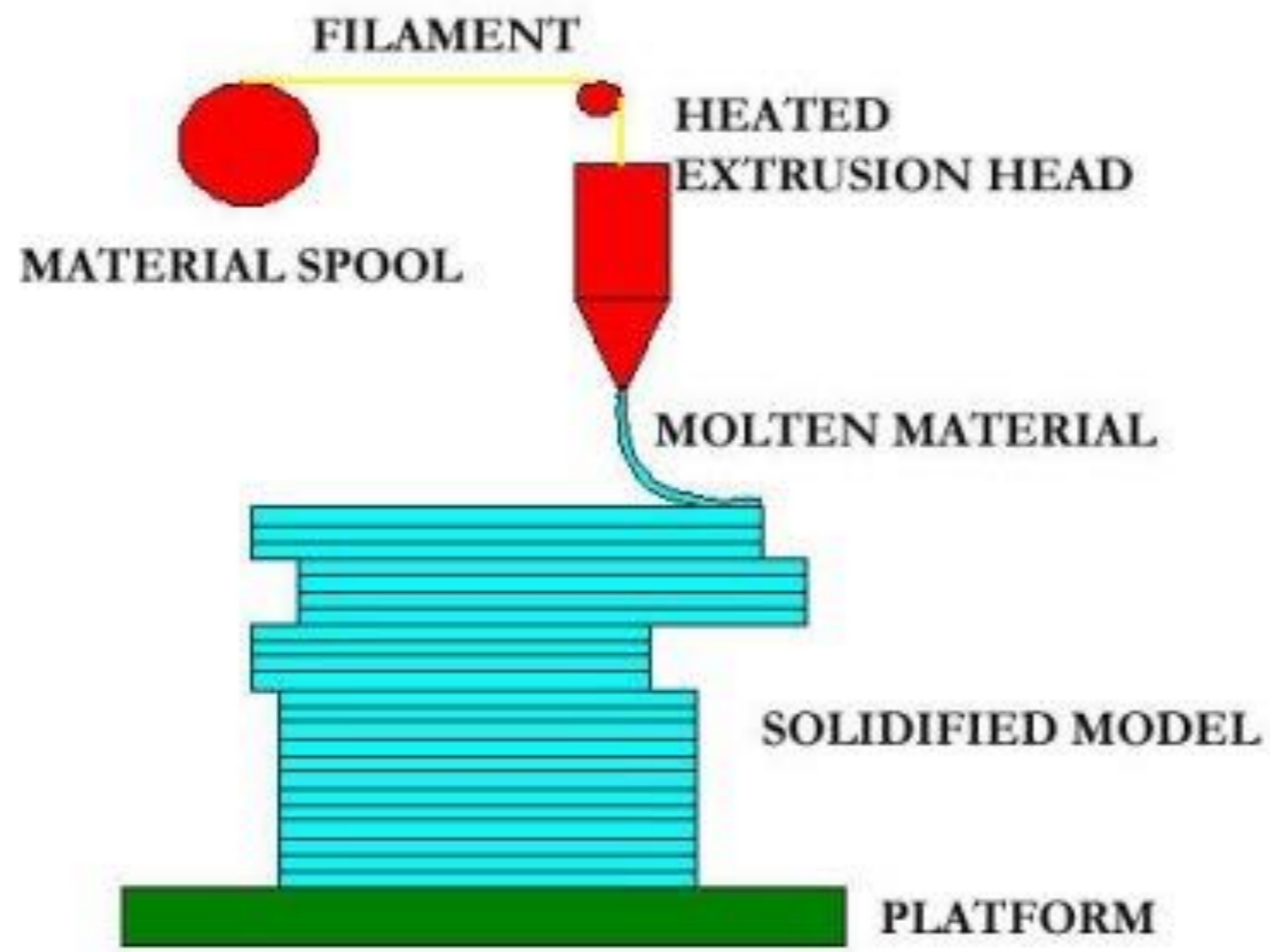
OpenSCAD

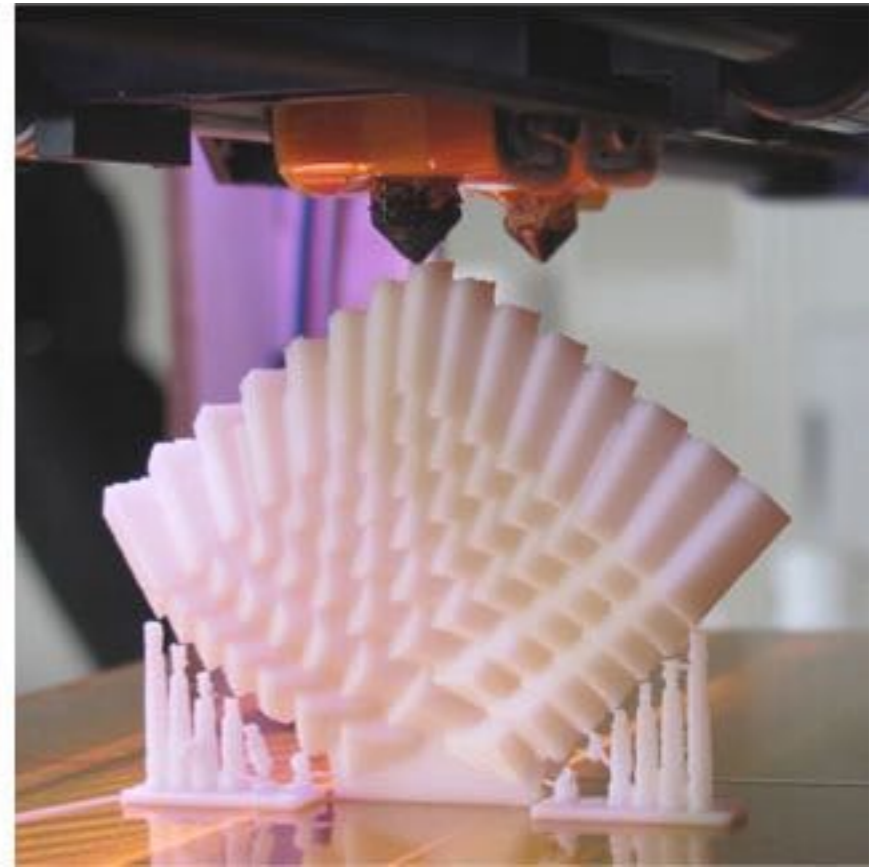
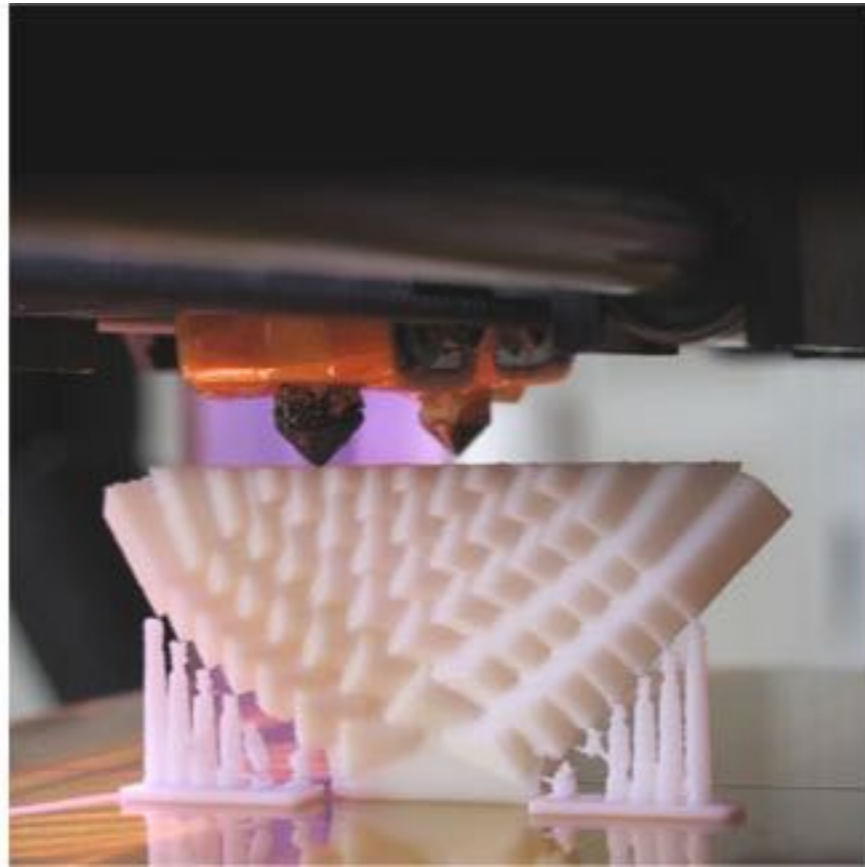
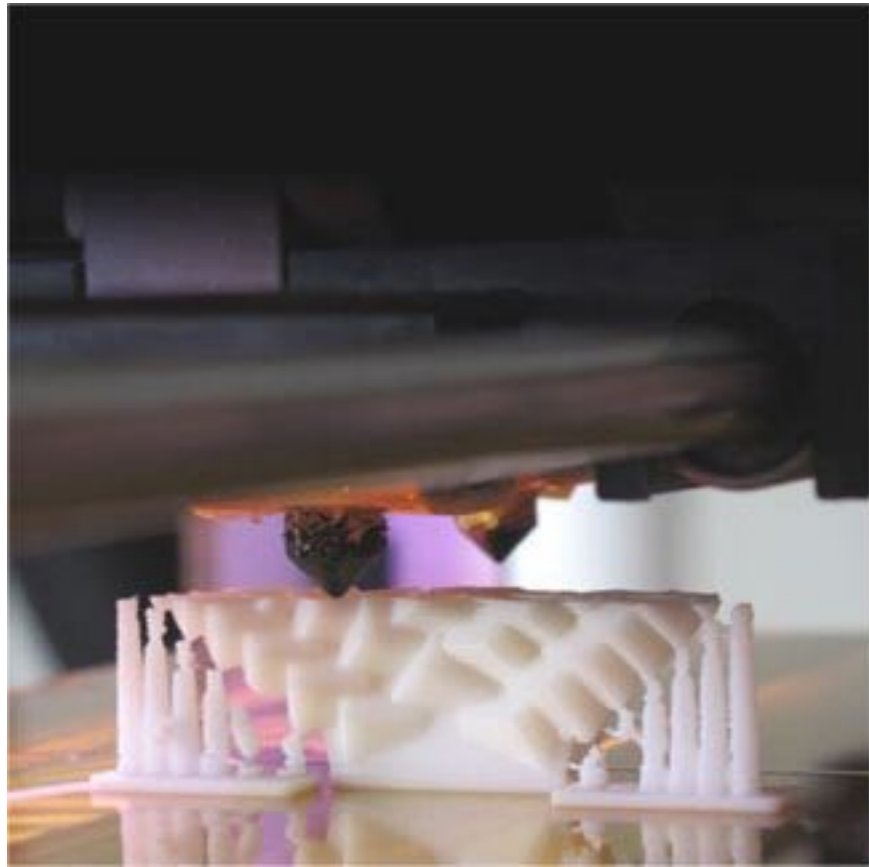
MARIUS KINTEL

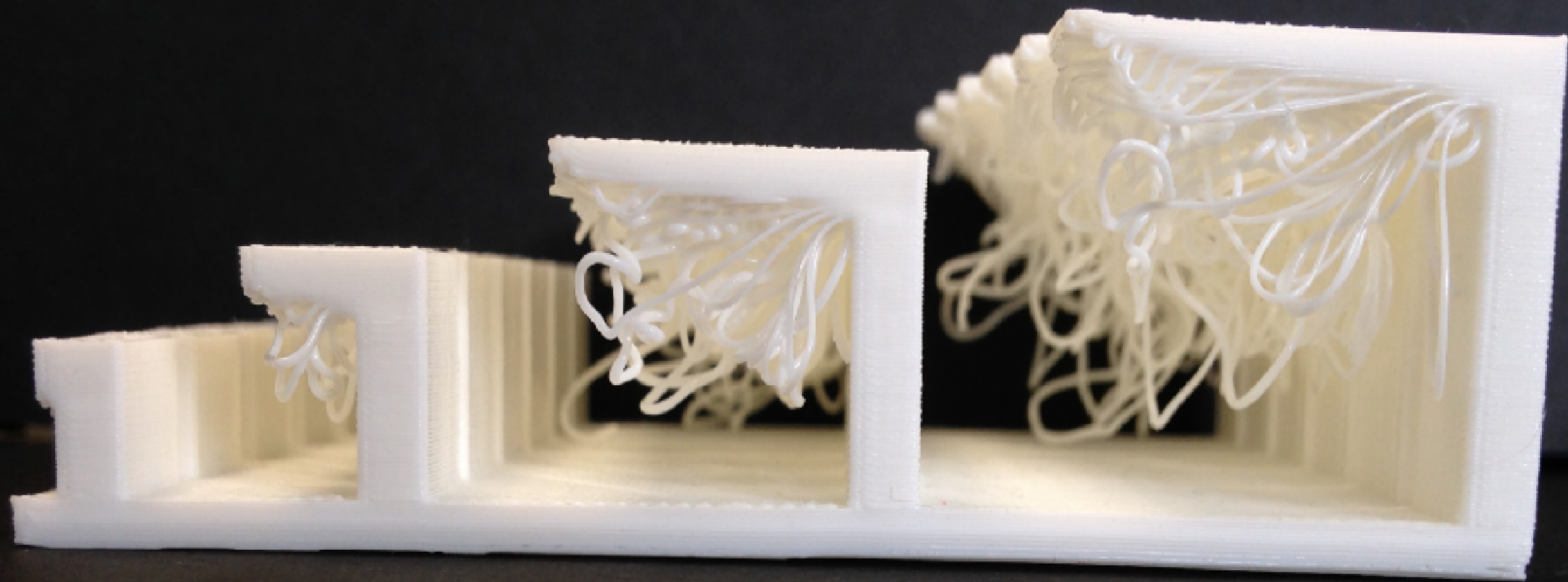
OUTLINE

- 3D Printing Primer
- OpenSCAD Walkthrough
- Hands-on tutorial
- Bring your own project

3D PRINTING











0%



5%



10%



15%



100%



75%



50%



25%



▲
Linear



▲
Hexagonal



▲
Morroccanstar

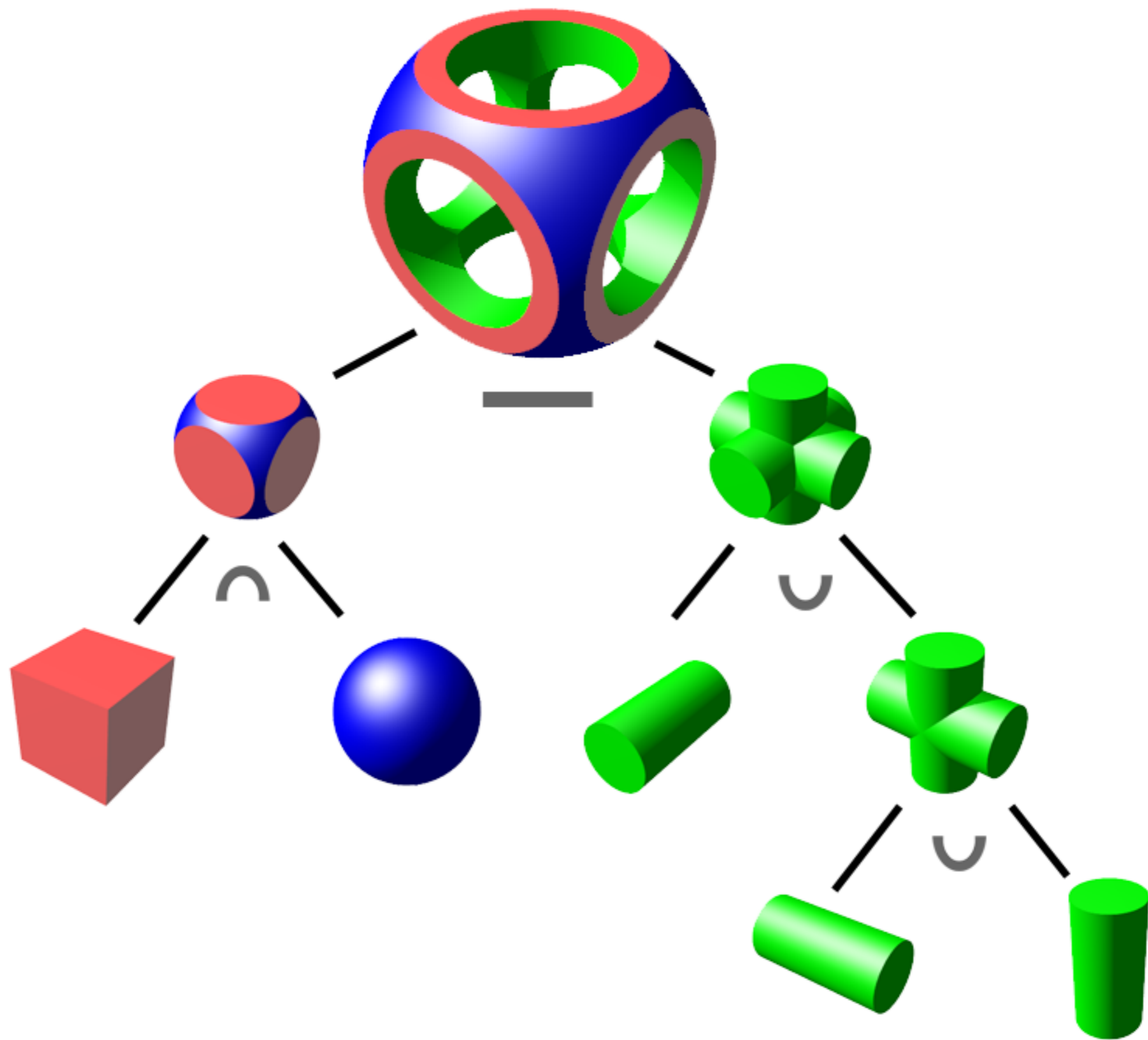


▲
Catfill

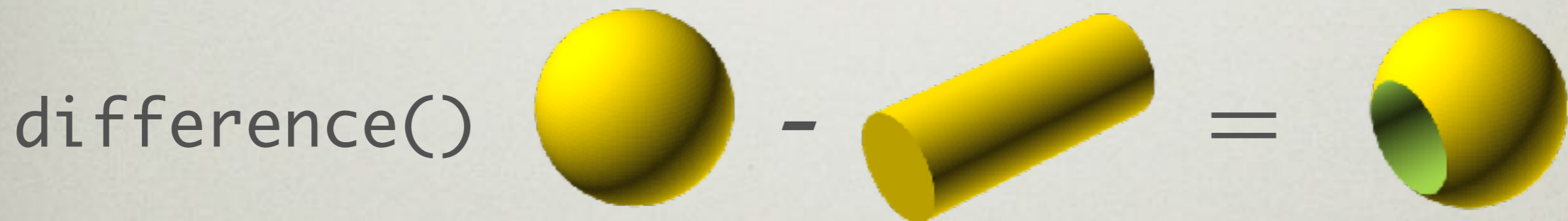
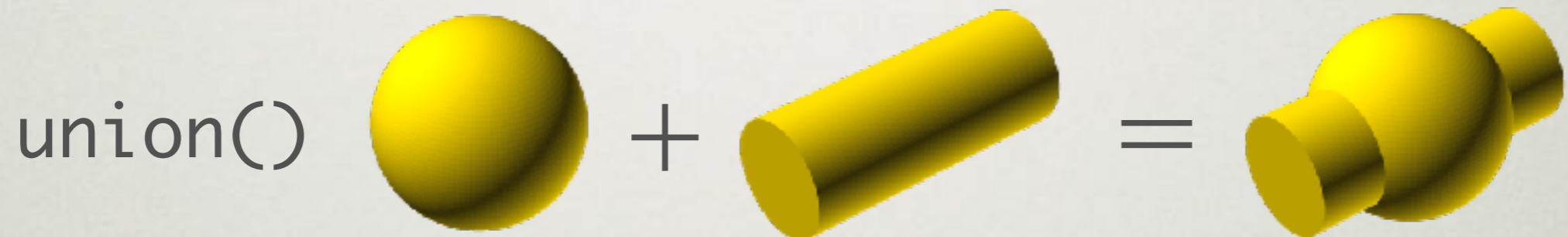
SOLID MODELING

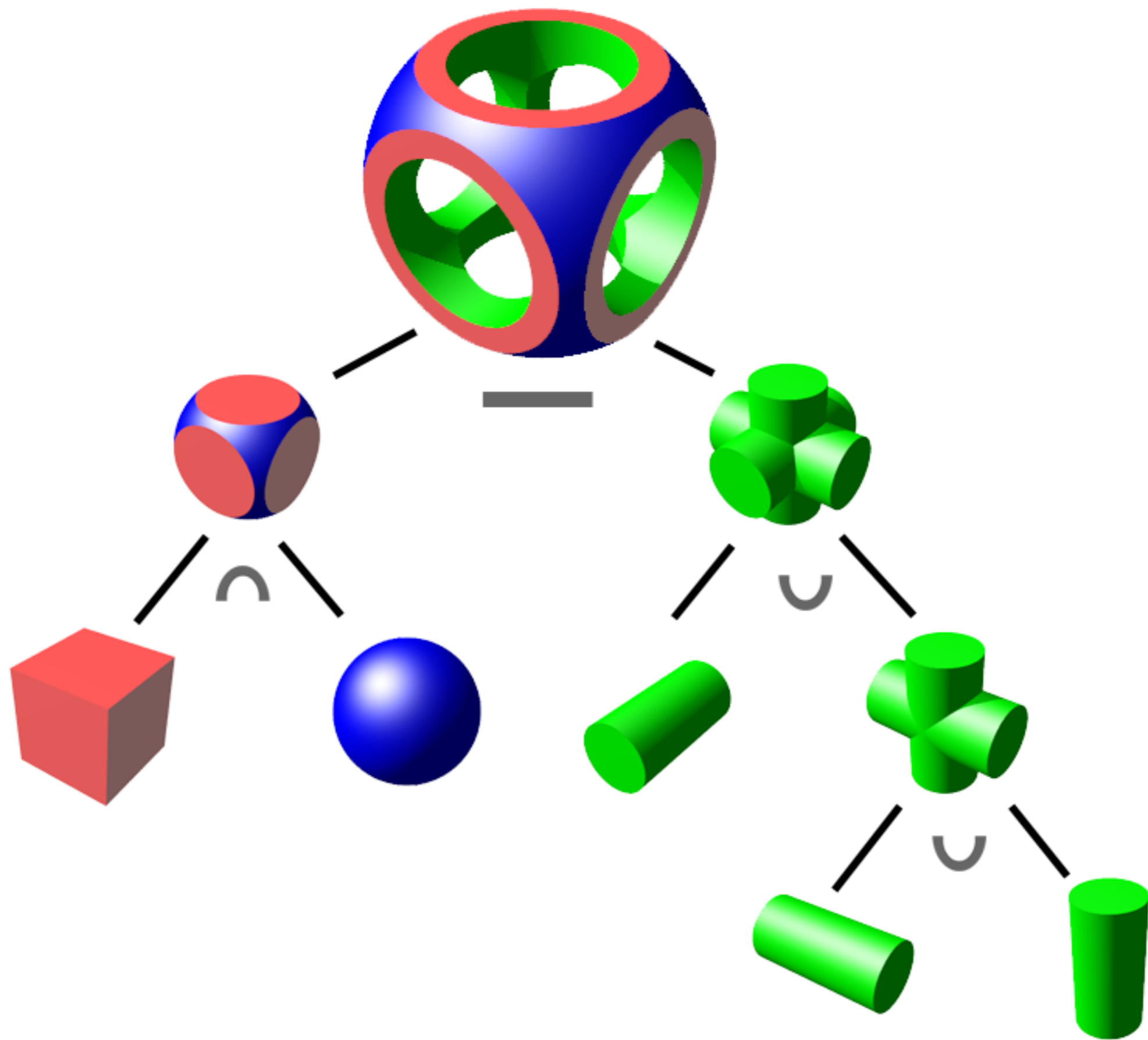
CSG

CONSTRUCTIVE SOLID GEOMETRY

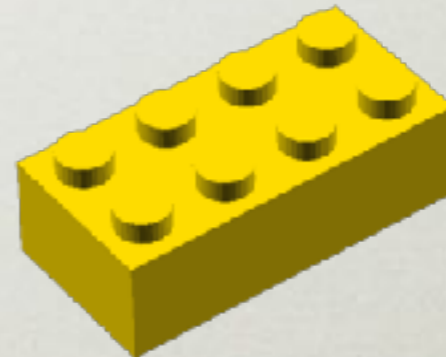


TEXTUAL DESCRIPTION

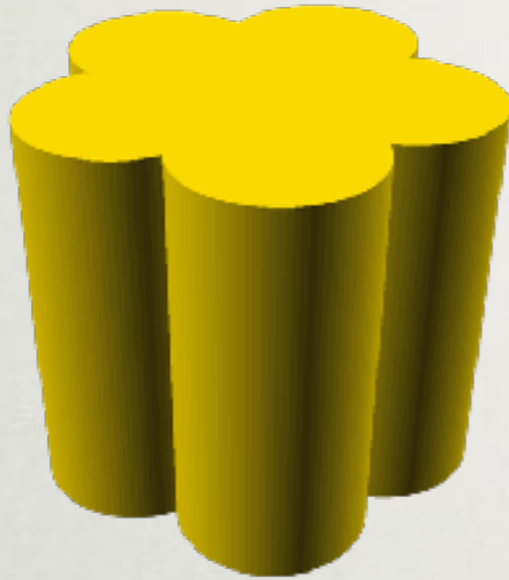




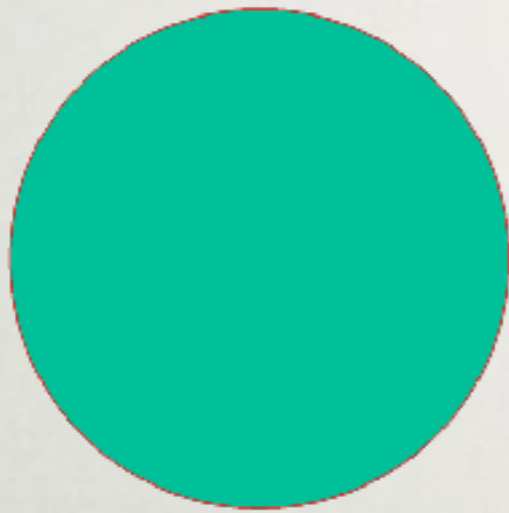
3D PRIMITIVES



EXTRUSIONS



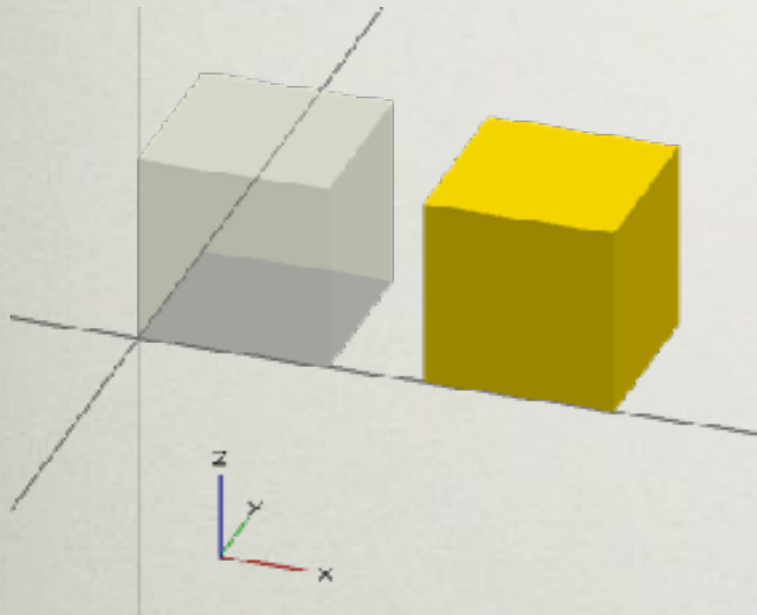
2D PRIMITIVES



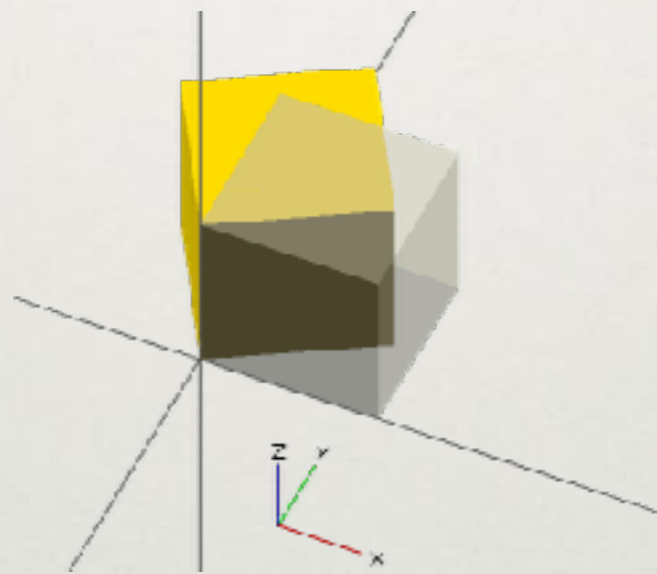
Sfære

TRANSFORMATIONS

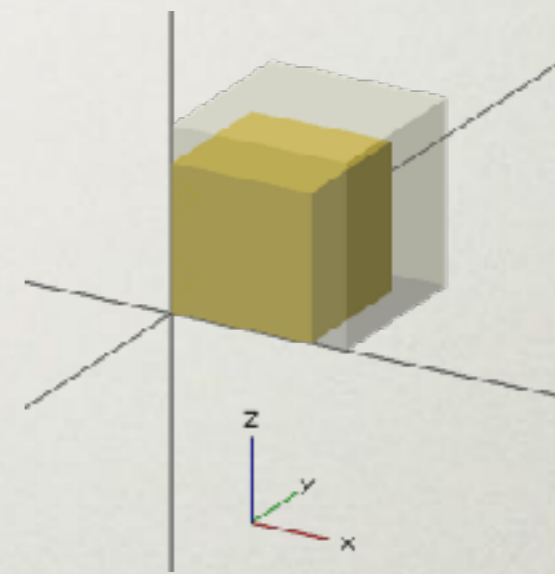
translate



rotate



scale



CUSTOMIZER

openscad.org / cheatsheet

OpenSCAD CheatSheet v2015.03

Syntax

```
var = value;  
module name(...) { ... }  
name();  
function name(...) = ...  
name();  
include <...scad>  
use <...scad>
```

2D

```
circle(radius | d=diameter)  
square(size,center)  
square([width,height],center)  
polygon([points])  
polygon([points],[paths])  
text(t, size, font,  
    halign, valign, spacing,  
    direction, language, script)
```

3D

```
sphere(radius | d=diameter)  
cube(size, center)  
cube([width,depth,height], center)  
cylinder(h,r|d,center)  
cylinder(h,r1|d1,r2|d2,center)  
polyhedron(points, triangles, convexity)
```

Transformations

```
translate([x,y,z])  
rotate([x,y,z])  
scale([x,y,z])  
resize([x,y,z],auto)  
mirror([x,y,z])  
multmatrix(m)  
color("colorname")  
color([r,g,b,a])  
offset(r|delta,chamfer)  
hull()  
minkowski()
```

Boolean operations

```
union()  
difference()  
intersection()
```

Modifier Characters

```
*      disable  
_      show only  
#      highlight / debug  
%      transparent / background
```

Mathematical

```
abs  
sign  
sin  
cos  
tan  
acos  
asin  
atan  
atan2  
floor  
round  
ceil  
ln  
len  
let  
log  
pow  
sqrt  
exp  
rands  
min  
max
```

Functions

```
concat  
lookup  
str  
chr  
search  
version  
version_num  
norm  
cross  
parent_module(idx)
```

Other

```
echo(...)  
for (i = [start:end]) { ... }  
for (i = [start:step:end]) { ... }  
for (i = [...,-,-]) { ... }  
intersection_for(i = [start:end]) { ... }  
intersection_for(i = [start:step:end]) { ... }  
intersection_for(i = [...,-,-]) { ... }  
if (...) { ... }  
assign(...) { ... }  
import("*.stl")  
linear_extrude(height,center,convexity, twist,slices)  
rotate_extrude(angle,convexity)  
surface(file = "...dat",center,convexity)  
projection(cut)  
render(convexity)  
children([idx])
```

List Comprehensions

```
Generate [ for (i = range|list) i ]  
Conditions [ for (i = ...) if (condition(i)) i ]  
Assignments [ for (i = ...) let (assignments) a ]
```

Special variables

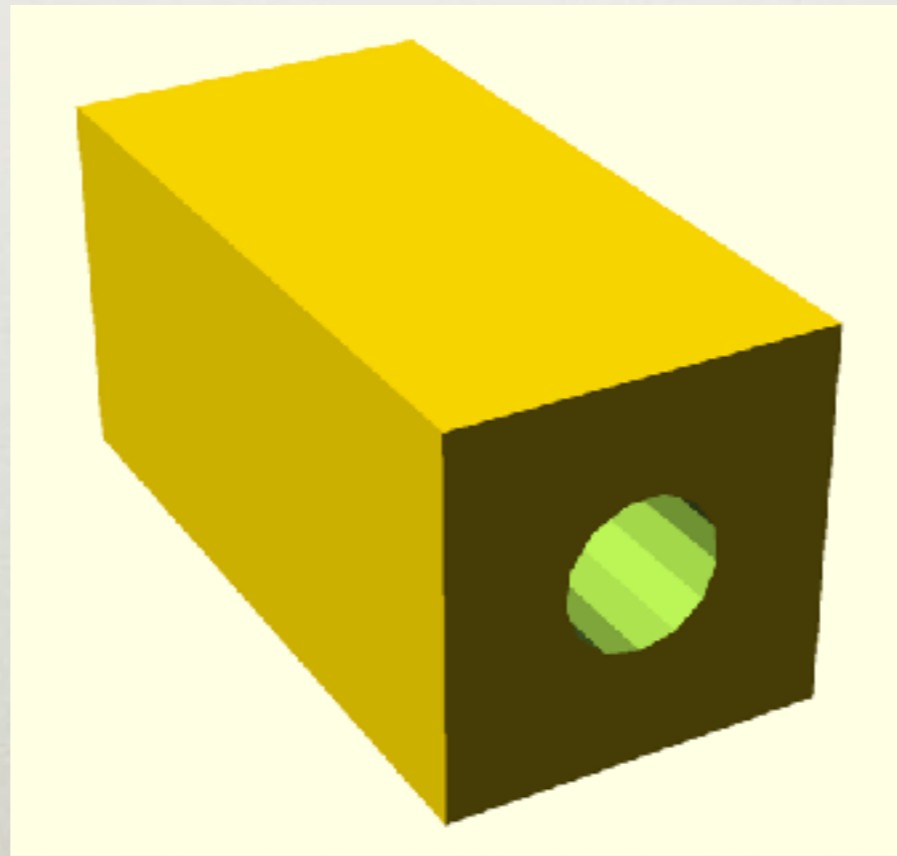
```
$fa  minimum angle  
$fs  minimum size  
$fn  number of fragments  
$t   animation step  
$vpr viewport rotation angles in degrees  
$vpt viewport translation  
$vpd viewport camera distance  
$children number of module children
```

VOCABULARY

- Syntax: Rules / grammar
- Parentheses: ()
- Square brackets: []
- Curly braces: { }
- Semicolon: ;

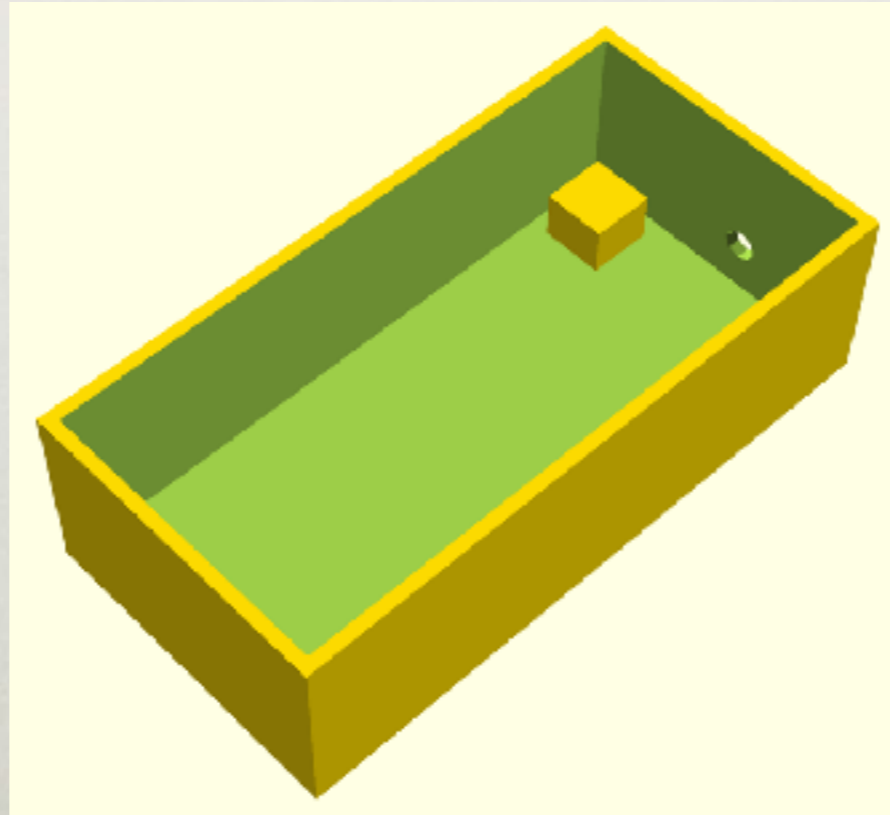
INTERACTIVE WALKTHROUGH

```
difference() {  
  cube([40,20,20]);  
  translate([-5,10,10]) rotate([0,90,0]) cylinder(h=50, r=4);  
}
```



```
difference() {  
  cube([100,50,30]);  
  translate([2,2,2]) cube([100-4,50-4,30]);  
  translate([-1,30,10]) rotate([0,90,0]) cylinder(h=10, d=5);  
}
```

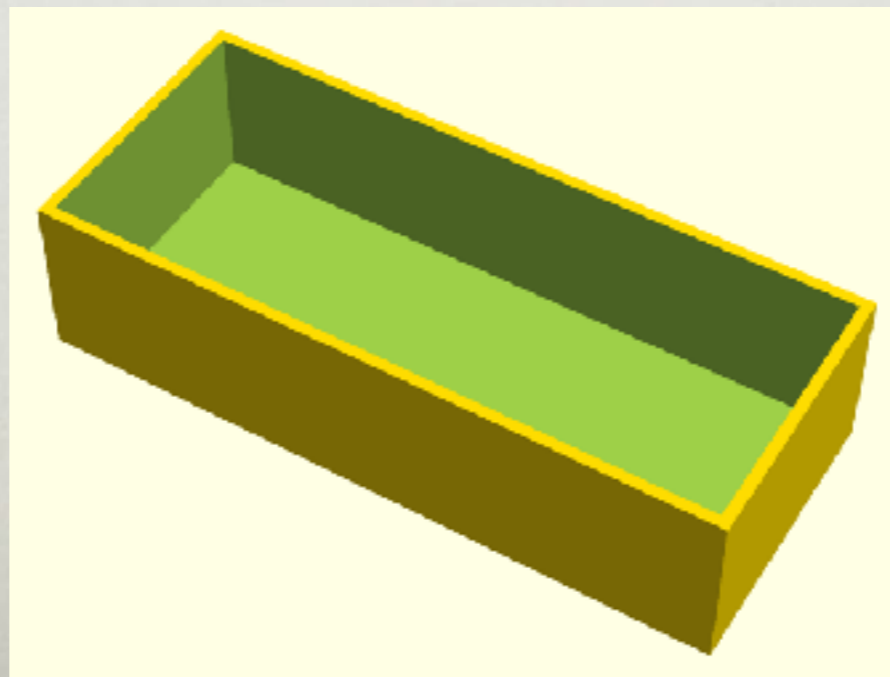
```
translate([5,5,0]) cube(10);
```

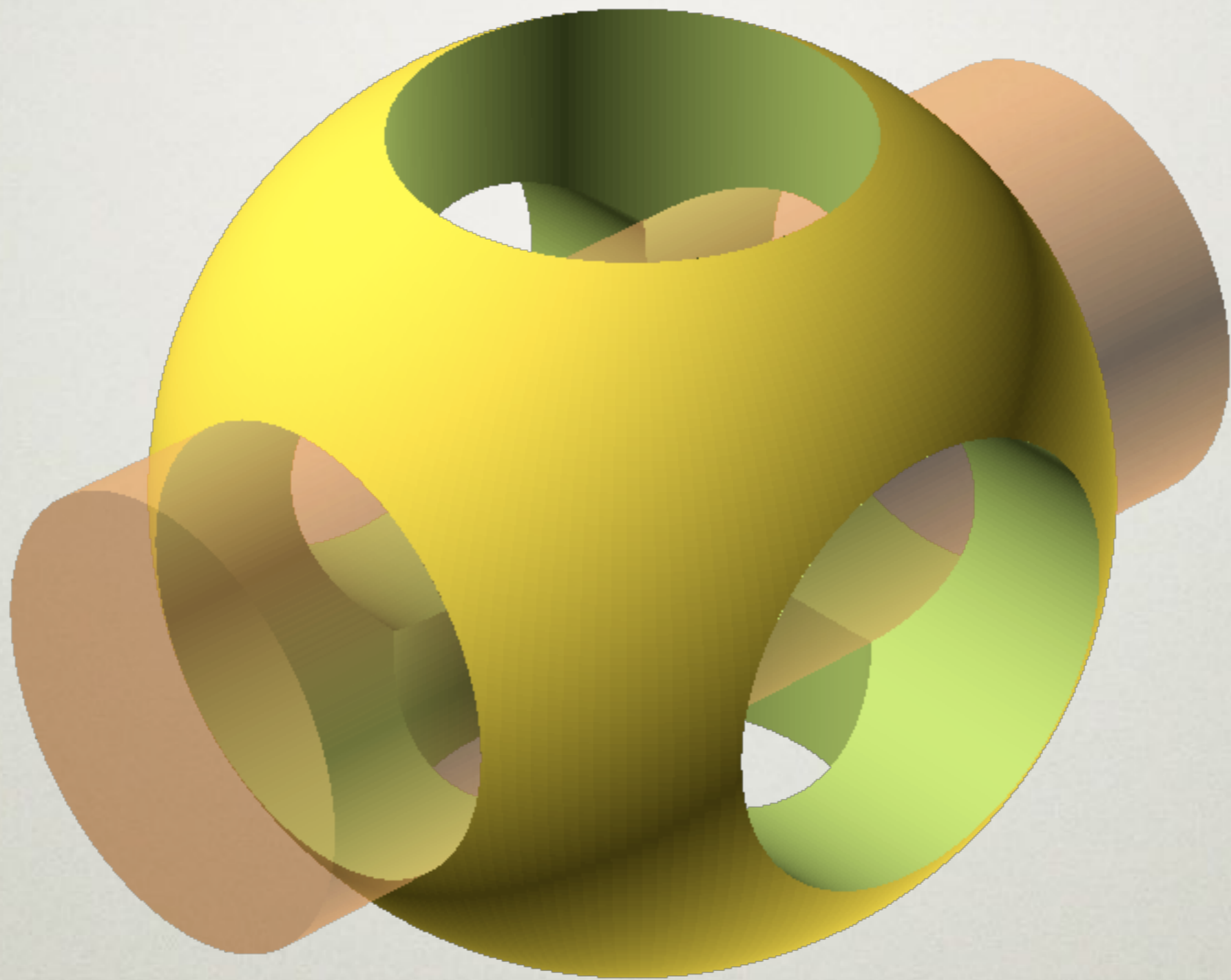


VARIABLES

```
Length = 120;  
Width = 50;  
Height = 30;  
Wall = 2;
```

```
difference() {  
    cube([Length, Width, Height]);  
    translate([Wall, Wall, Wall])  
        cube([Length-2*Wall, Width-2*Wall, Height]);  
}
```





TEXT

```
translate([20,0,0])  
  linear_extrude(height=10)  
    text("MARIUS", size=20);
```

```
difference() {  
  translate([0,2.5,0]) cube([120,15,3]);  
  translate([10,10,-1]) cylinder(h=20, d=10);  
}
```



```

translate([20,0,0])
  linear_extrude(height=10)
    text("MARIUS", size=20, font="Noteworthy:style=Bold");

difference() {
  union() {
    translate([7.5,2.5,0]) cube([120-7.5,15,3]);
    translate([7.5,10,0]) cylinder(d=15, h=3);
  }
  translate([10,10,-1]) cylinder(h=20, d=10);
}

```



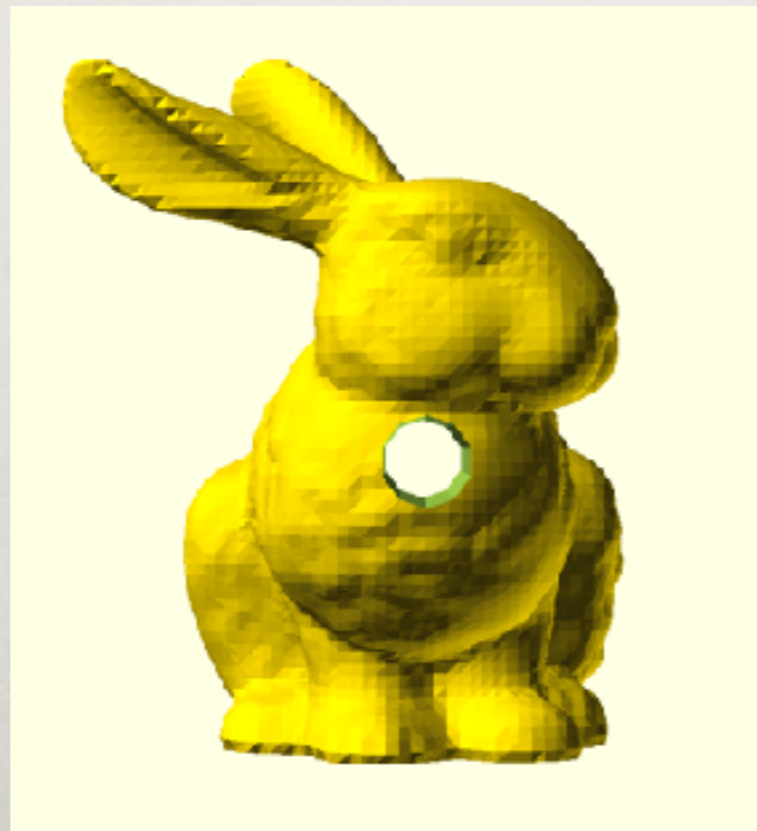
IMPORTING FILES

```
import("filename.stl");
```

```
linear_extrude(height=10) import("filename.dxf");
```

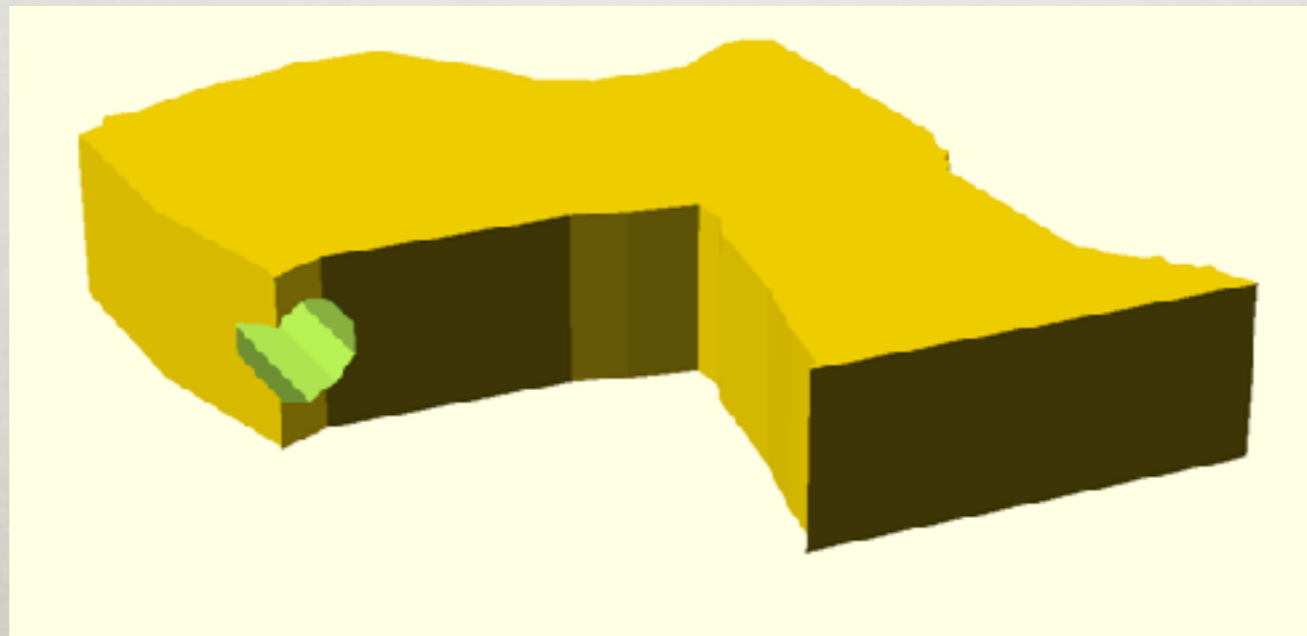
IMPORT STL

```
difference() {  
  import("bunny.stl", convexity=3);  
  translate([0,-8,40])  
  rotate([0,90,0]) cylinder(h=100, r=3, center=true);  
}
```



IMPORT DXF

```
difference() {  
  linear_extrude(height=10, convexity=3) import("drawing.dxf");  
  translate([0,60,5])  
  rotate([0,90,0])  
  cylinder(h=100, r=3, center=true);  
}
```



RESOURCES

- Docs, tutorials
 - openscad.org/documentation.html
- Examples
 - thingiverse.com/search?q=openscad
 - thingiverse.com/customizable