OpenSCAD Workshop

Written by Mark Webster

Summary: Learn to create 3D and 2D models in Openscad.

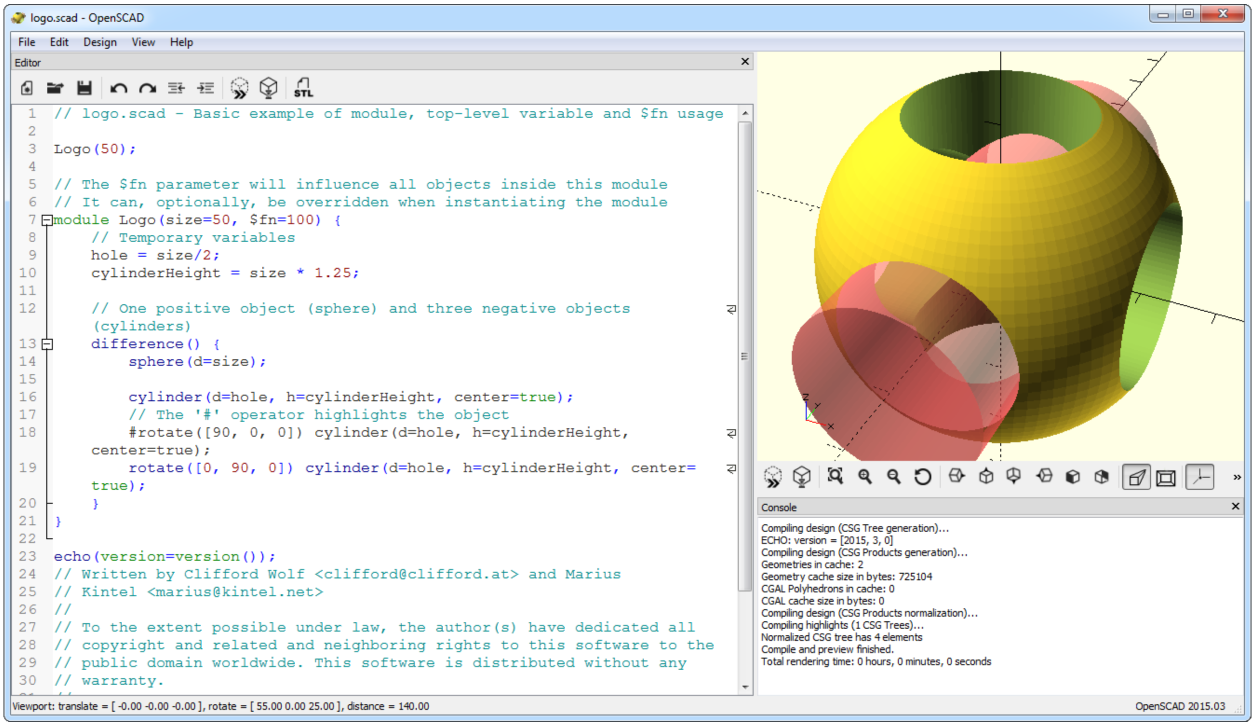
Introduction

OpenSCAD is a free software program available on Linux, MacOS, and Windows for creating 2D and 3D models that can be printed with 3D printers, CNC mills, or even laser cutters. The outputs from OpenSCAD are 3D or 2D format files.

OpenSCAD differs from graphical manipulation modelers such as Fusion 360 because OpenSCAD develops models using programmer friendly parametric text scripts. Basic shapes such as cylinders, spheres, boxes, polygons, or text are transformed or combined to produce the final model.

The script is written in a text editor window on the left hand side and the result is displayed in a graphical preview window on the right. The graphical rendering can be rotated, scaled, or shifted with a mouse. The resulting model can be exported in many formats such as STL, DXF, SVG or even PNG. The OpenSCAD files *.scad are generic text files and can be opened in any text editor.

Information about the rendered file is displayed in the lower right window. Debugging information from the OpenSCAD program can also be displayed in that window.



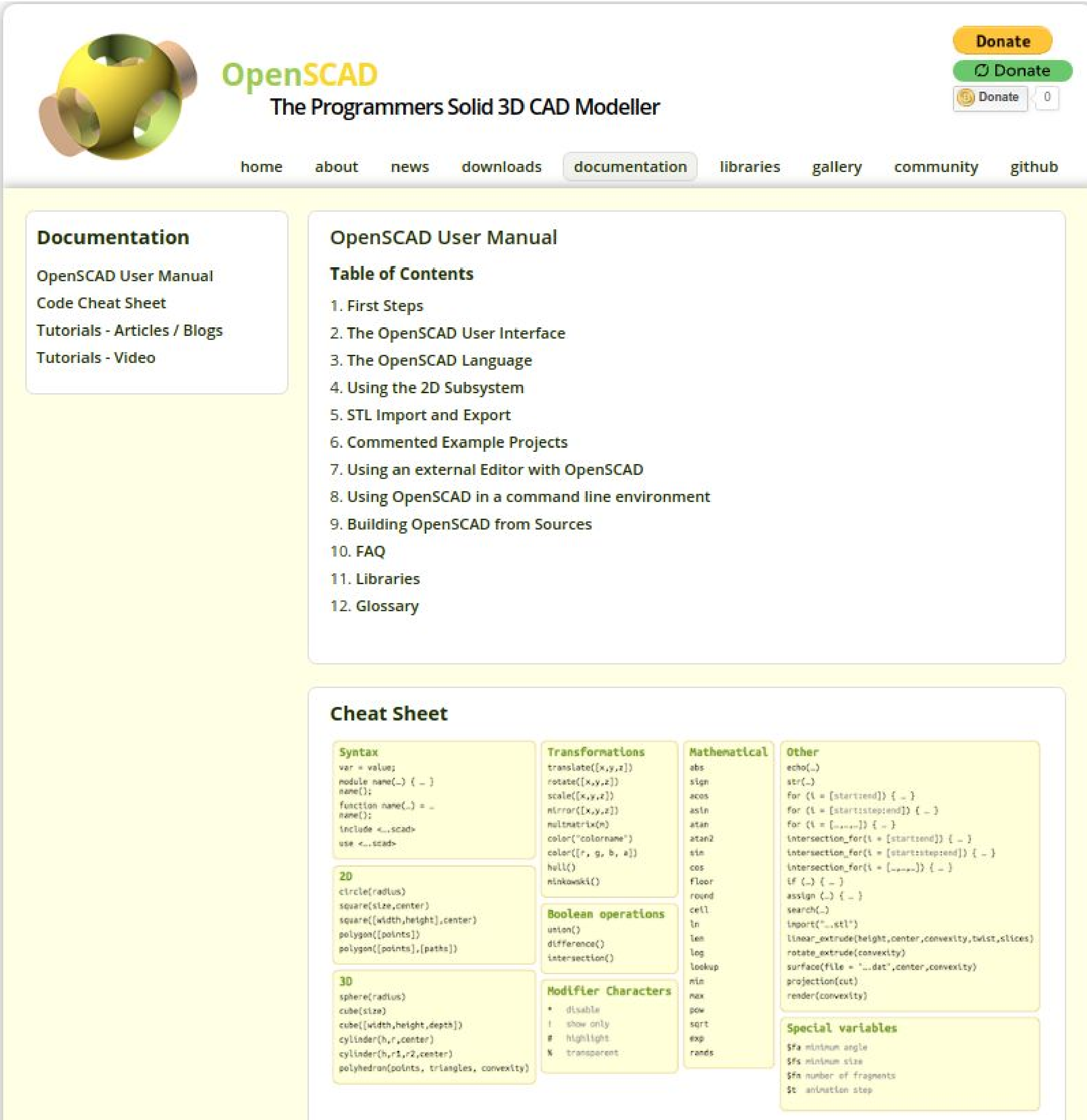
(Image from Wikipedia)

Installation & Documentation

All versions can be downloaded from the project website: https://www.openscad.org/

Installation is simple, although rendering speed will depend on the GPU on the computer and the drivers installed.

Extensive tutorials and documentation exists on the www.openscad.org website and from the Help menu in the program.



The OpenSCAD cheat sheet is the most useful in general. This cheat sheet is available on the Help menu and online. On the cheat sheet just click on a function to jump to more extensive documentation.

Many Youtube tutorials exist online, and example scad files can be found in Thingiverse.

3D Primitives

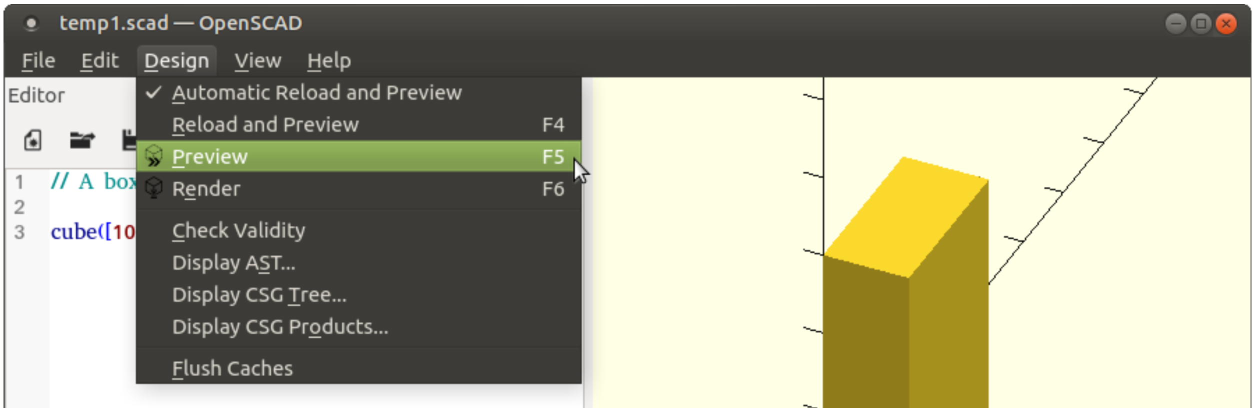
```
3D
sphere(radius | d=diameter)
cube(size, center)
cube([width,depth,height], center)
cylinder(h,r|d,center)
cylinder(h,r1|d1,r2|d2,center)
polyhedron(points, faces, convexity)
import("….ext")
linear_extrude(height,center,convexity,twist,slices)
rotate_extrude(angle,convexity)
surface(file = "….ext",center,convexity)
```

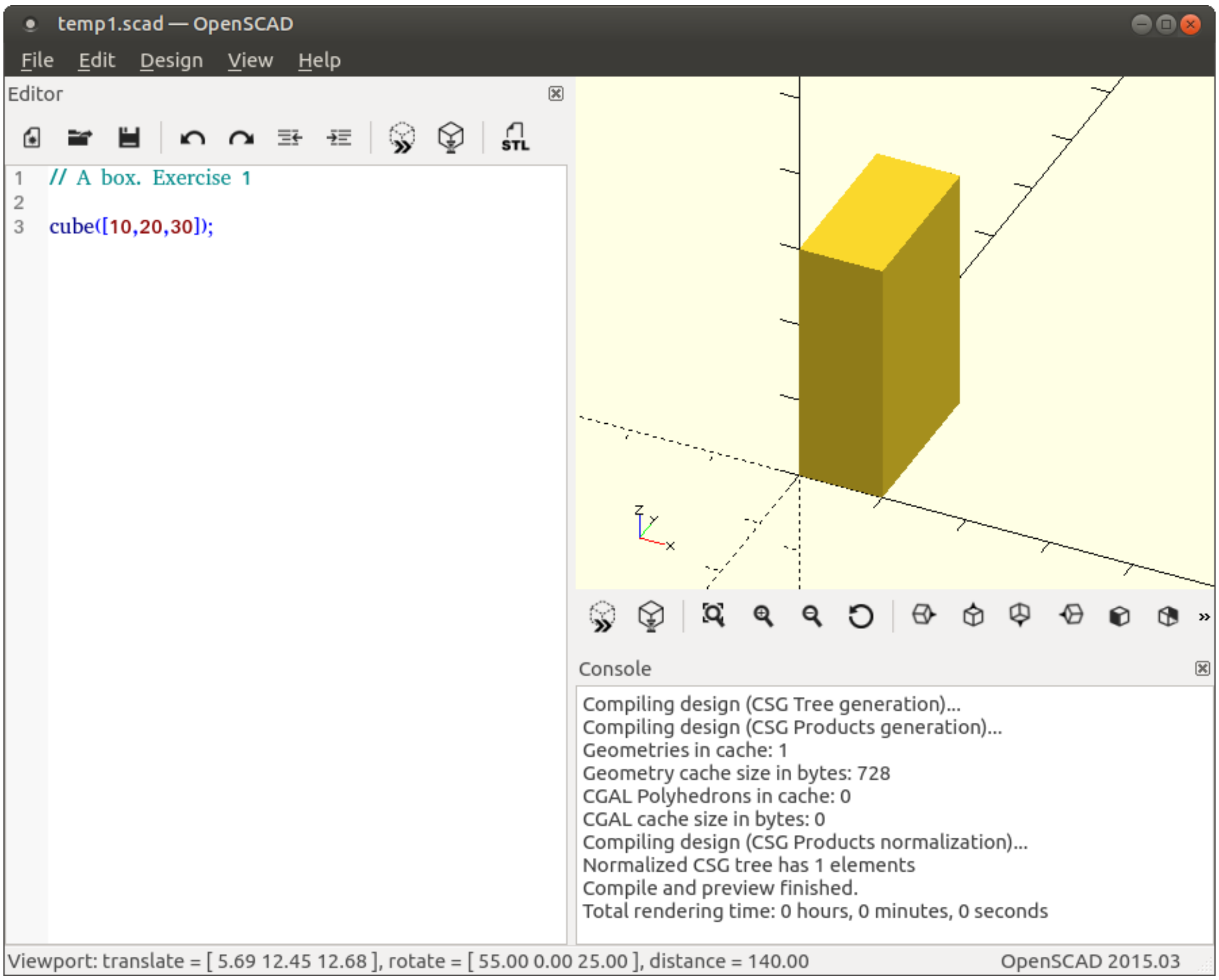
Exercise 1: Make a box

Launch the software and type in the script window
cube([10,20,30]);
The x,y, z dimensions of the box (cube) are specified as a tuple [x,y,z]
The name “cube” is not exactly right since width, length, and height can all be different.

OpenSCAD uses C style syntax. Single line comments are C style begine with a //. All lines end with a semicolon ;.

Save the file as a temporary file somewhere on the local computer. Each time the file is saved a preview is generated. A preview can be forced from the Design menu or the Preview icon. A more refined view can be created by selecting Render or clicking the Render icon.



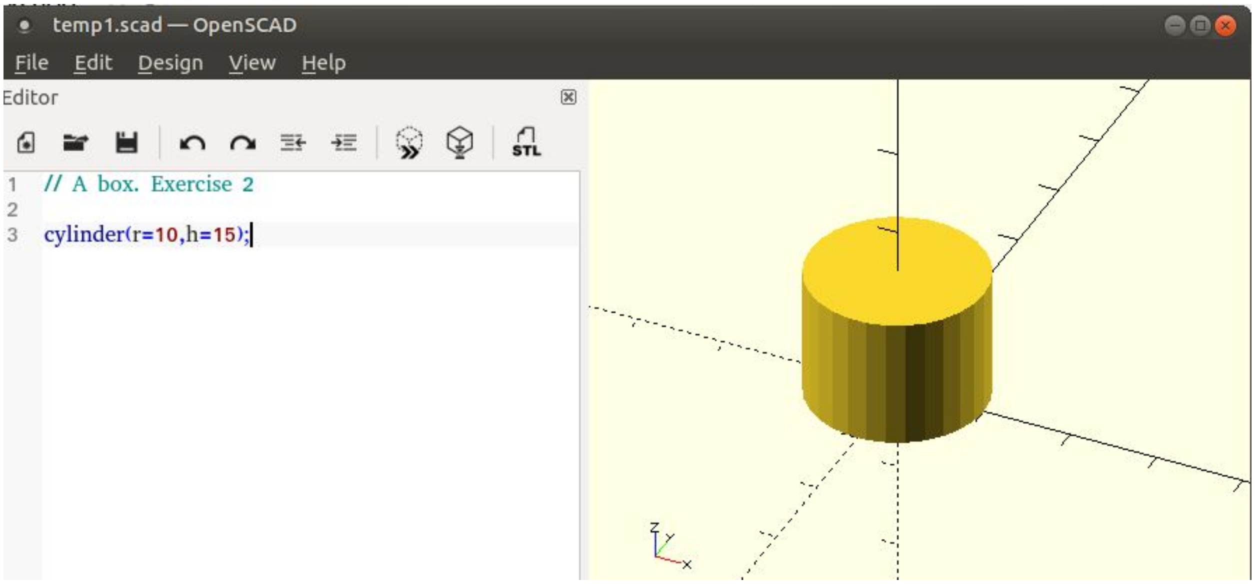


Rotate the image by clicking in the preview with the left mouse button and drag. Translate the image with the right mouse button and drag. The mouse wheel will zoom in or out.

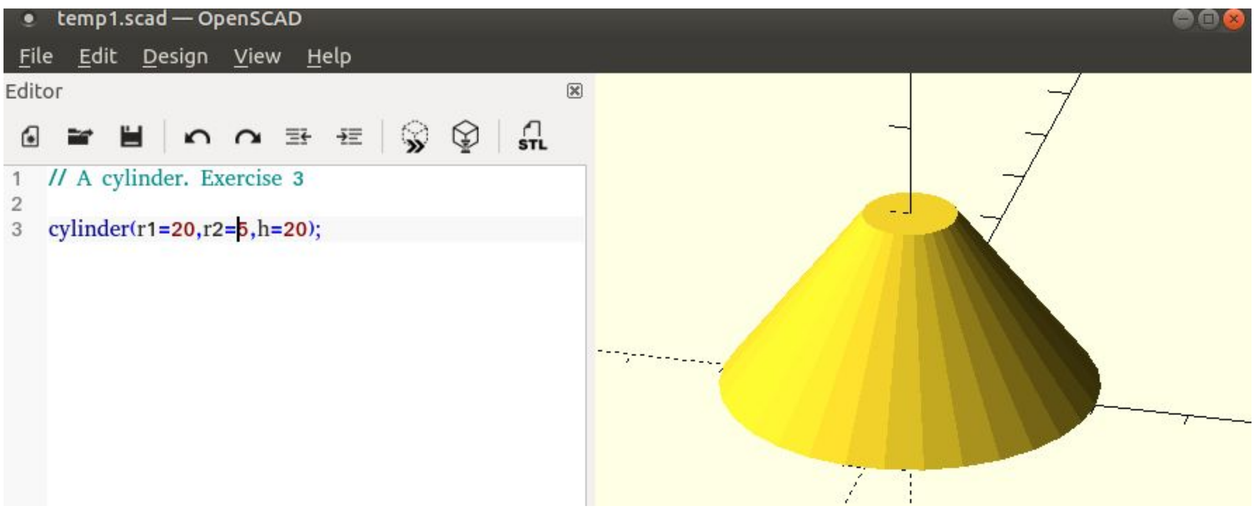
Exercise 2: Make a cylinder

In the script window type

cylinder(r=10,h=20);

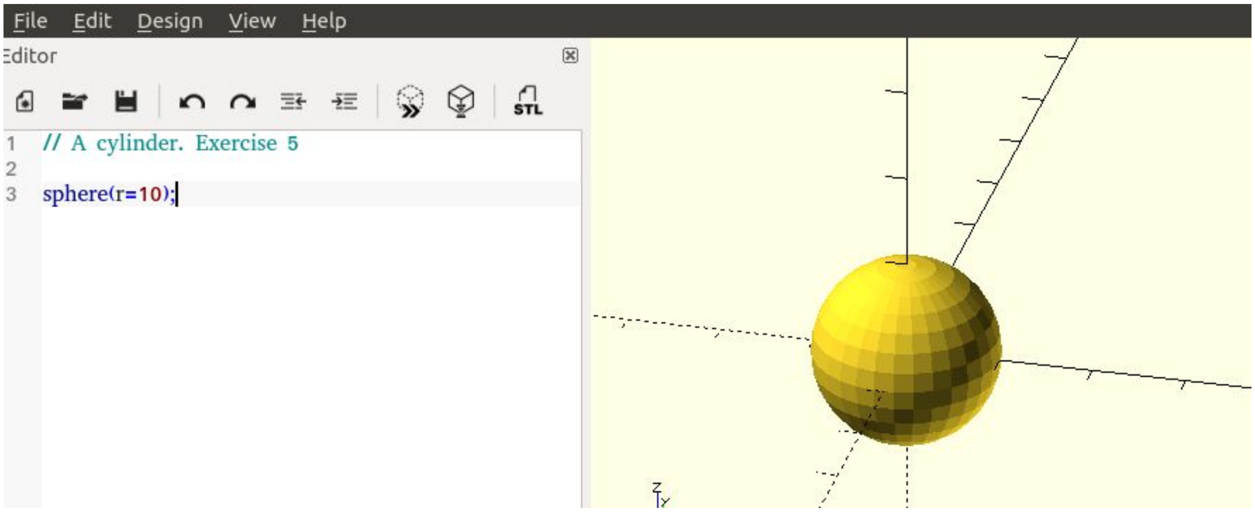


A cone is made by defining a different radius at the bottom and top of the cylinder.



Exercise 3: Sphere

A sphere is made with the command:
sphere(r=10);

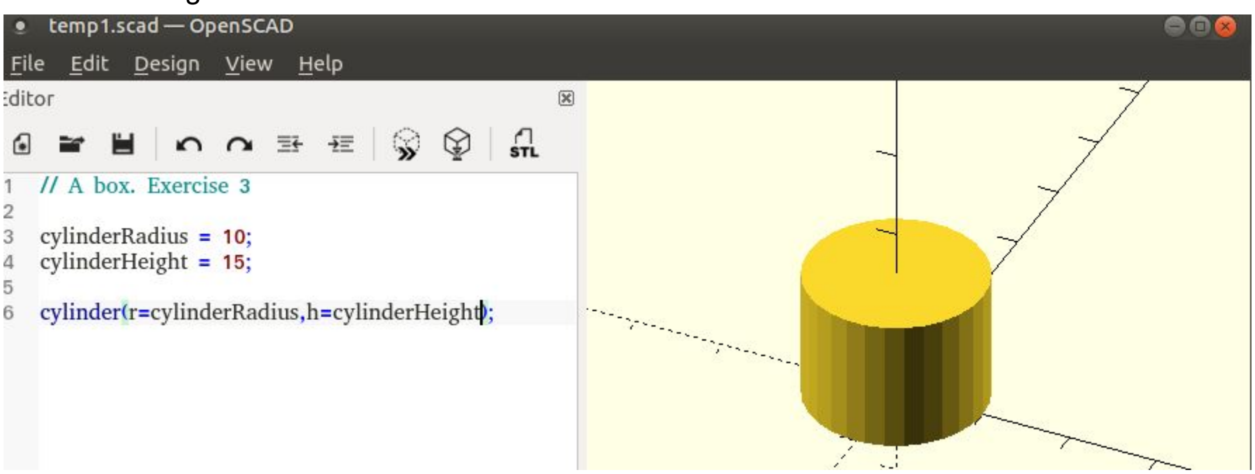


Exercise 4: Variables

The power of a parametric modeler is to specify all dimensions as variables. Then the value can be changed in one spot and the entire model will be modified.

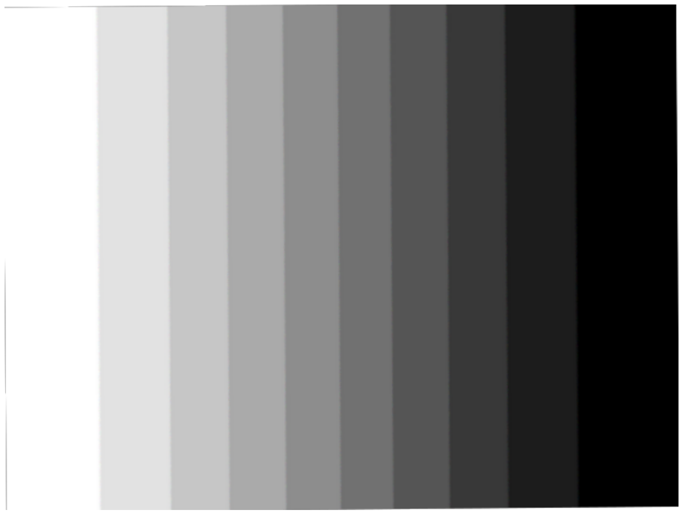
Type
cylinderRadius = 10;
cylinderHeight = 15;
cylinder(r=cylinderRadius,h=cylinderHeight);

The same image as before should be created.



Exercise 5: Surfaces (Graphics or data files)

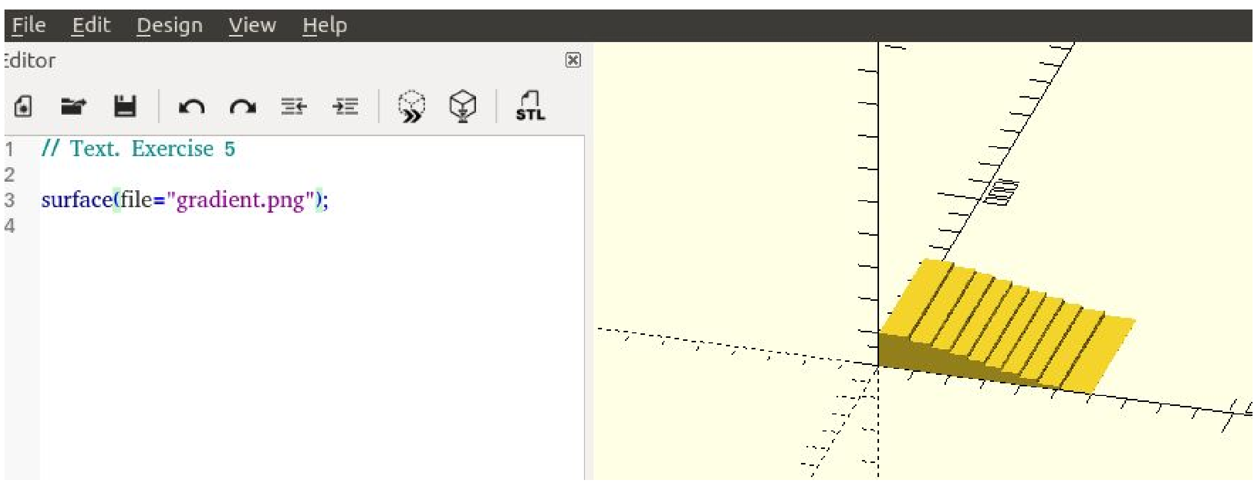
Find and save some simple graphics file in PNG format. Grayscale is simpler to understand. Or create a matrix data file and save as a DAT text file. This example file is called “gradient.png”



Type in the editor window:

surface(file=”gradient.png”);

The result is a surface where the shade of gray corresponds to a height



A data file would be a text file that looks like:

```
#surface.dat
10 9 8 7 6 5 5 5 5 5
9 8 7 6 6 4 3 2 1 0
8 7 6 6 4 3 2 1 0 0
7 6 6 4 3 2 1 0 0 0
6 6 4 3 2 1 1 0 0 0
6 6 3 2 1 1 1 0 0 0
6 6 2 1 1 1 1 0 0 0
6 6 1 0 0 0 0 0 0 0
3 1 0 0 0 0 0 0 0 0
```

3 0 0 0 0 0 0 0 0 0

OpenSCAD can be used to create a 3D image of datasets from engineering or science experiments. Also, graphical pictures like logos can be inserted into a 3D model by using the surface() function.

2D Shapes

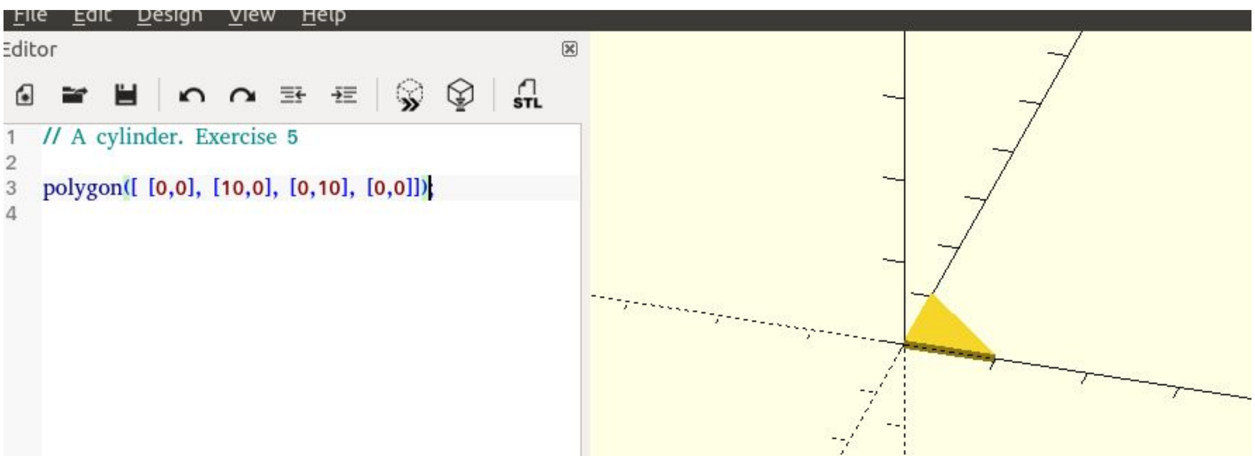
Two dimensional shapes can be drawn, and then extruded into 3 dimensions if desired. Circle is like sphere, square is like cube, polygon takes a list of points.

Exercise 6: polygon

In the script window type the line, then save or click preview:

```
polygon([ [0,0], [10,0], [0,10], [0,0]]) ;
```

Notice a list of points is [[x1,y1], [x2,y2], [x3,y3]...]



Exercise 7: Extrude a 2D shape

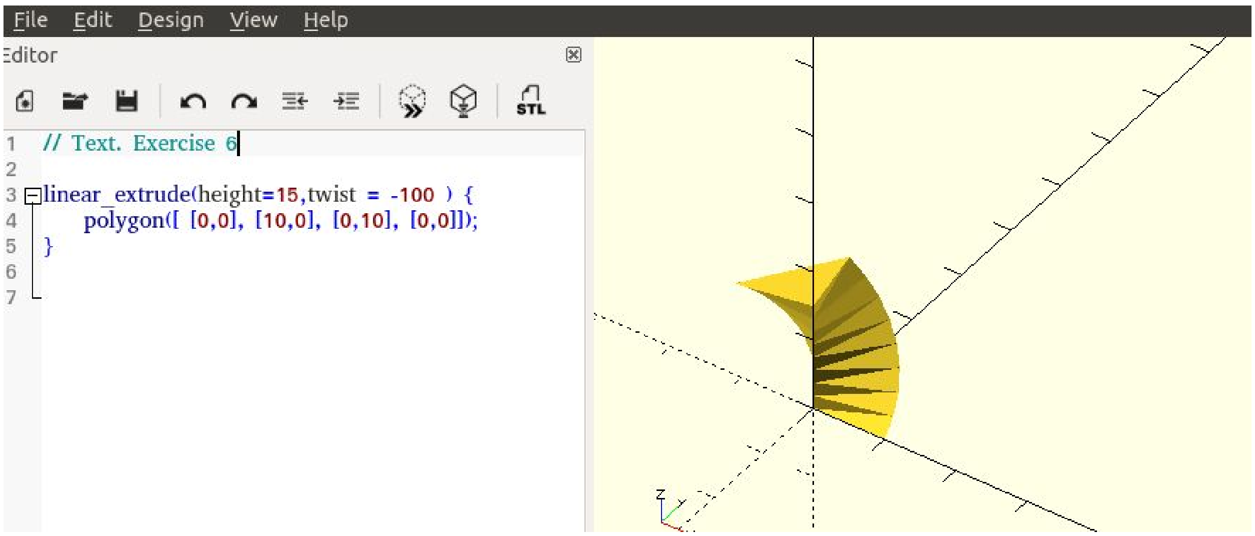
A two dimensional shape can be extruded or stretched into 3 dimensions with the linear_extrude(0) or rotate_extrude() commands.

Like C, multiple lines within a function are contained within curly braces {, }. Traditionally lines within a function are indented for human readability.

If no twisting is desired while extruding, then just leave out the twist parameter or set twist = 0.

Type into the script window:

```
linear_extrude(height=15,twist = -100 ) {
   polygon([ [0,0], [10,0], [0,10], [0,0]]);
}
```

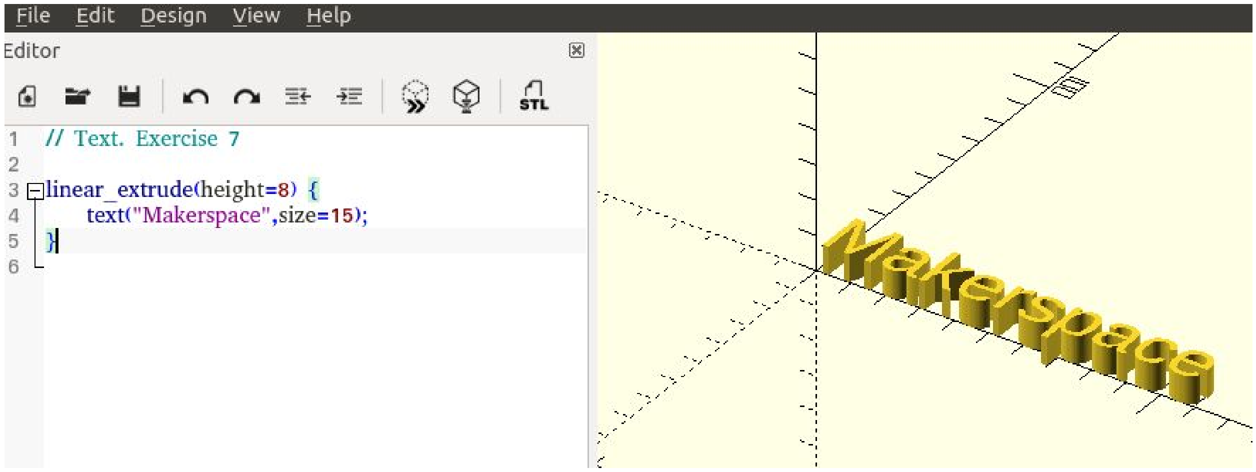


Exercise 8: Text

Text can be created in 2 dimensions and extruded into 3D if desired.
Type:

```
linear_extrude(height=8) {
   text("Makerspace",size=15);
}
```

There are many options for text like font, centering, etc, so please consult the documentation for details.



Modifying and Combining Shapes

Objects in OpenSCAD can be translated, rotated, and multiple shapes can be combined, or subtracted from each other.

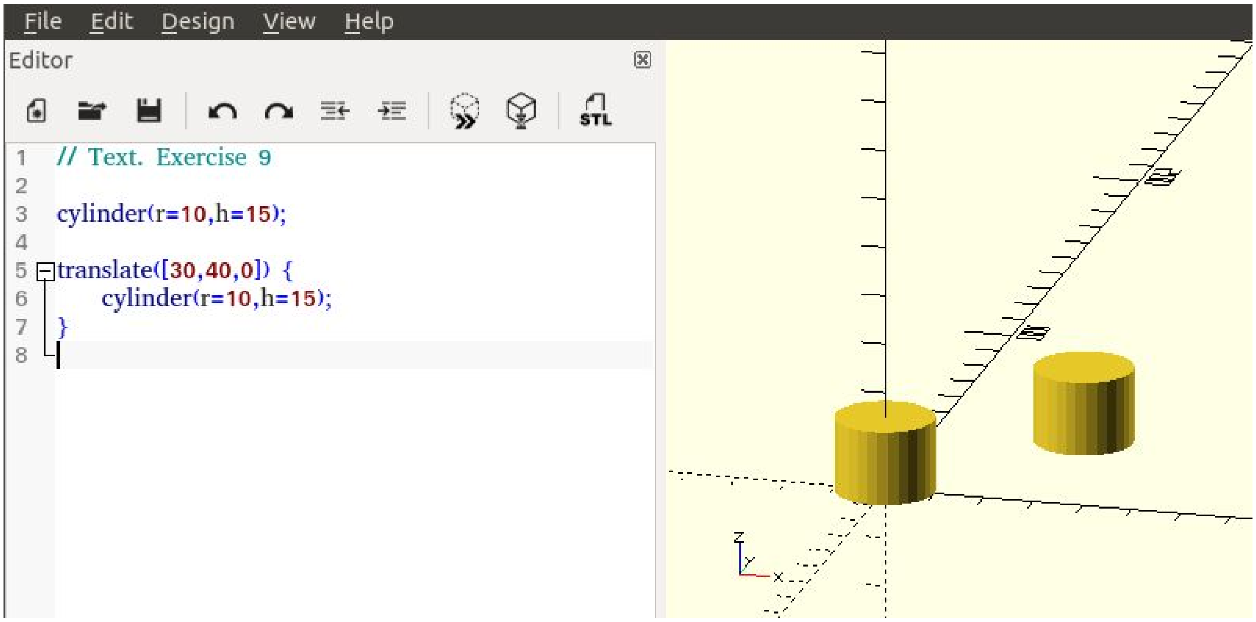
Example 9: Translate

The translate function shifts an object x,y,z units. OpenSCAD defines the direction of the translation by a vector [x,y,z]. The translation is from the origin [0,0,0].

Type into the editor

```
cylinder(r=10,h=15);

translate([30,40,0]) {
   cylinder(r=10,h=15);
}
```

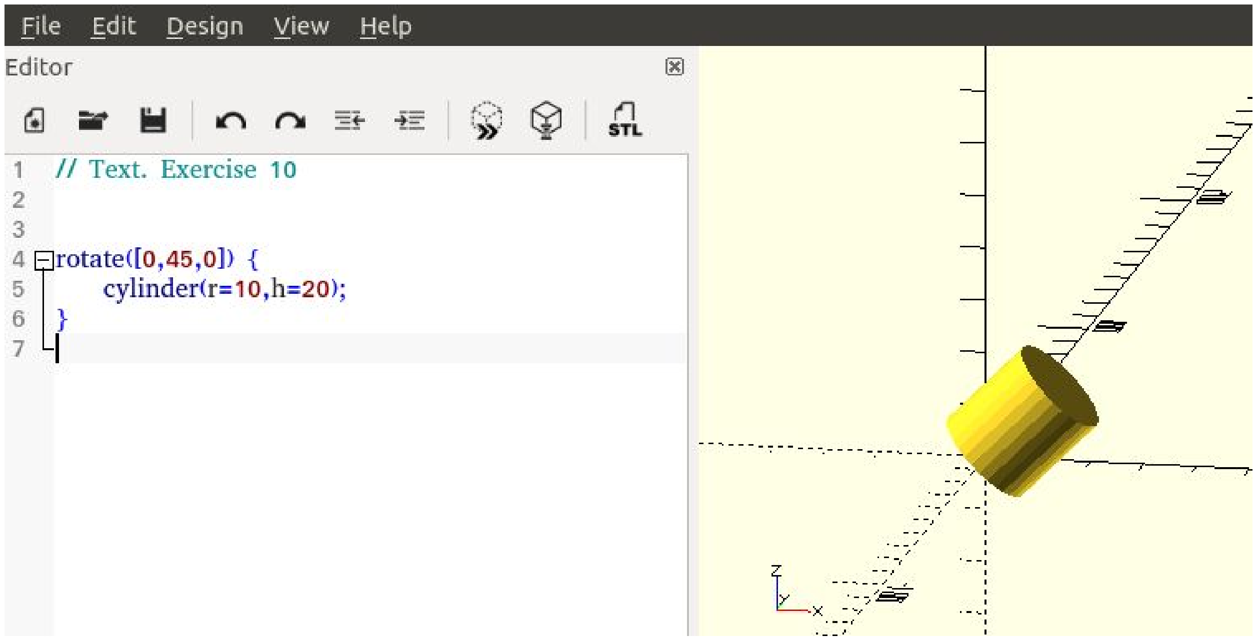


Exercise 10: Rotate

Type into the editor window:

```
rotate([0,45,0]) {
   cylinder(r=10,h=20);
}
```

This rotates the object 45 degrees around the around the y axis. The rotation is defined as degrees around a certain axis. The three axes are <x,y,z>. The 45 in the y position rotates the object around the y axis.



Exercise 11: Difference

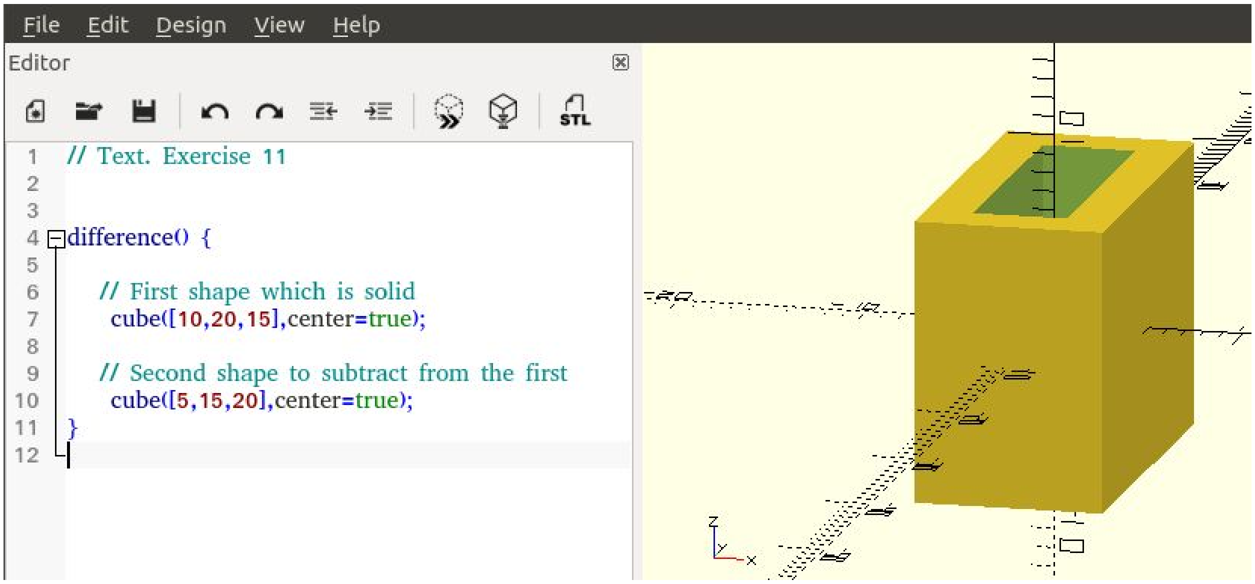
Subtract a second shape from the first shape. In this example, subtract the inside of a box from the outside of the box.

Type into the editor window:

```
difference() {

   // First shape which is solid
    cube([10,20,15],center=true);

   // Second shape to subtract from the first
    cube([5,15,20],center=true);
}
```



Other transformations are scaling, mirroring, coloring, and more obscure but useful ones like hull or minkowski.

To combine several shapes use the “union() { }” function. A generic OpenSCAD program might be:

```
difference() {
    // Positive shapes to keep
    union() {
    }
    // Shapes to subtract form the positive volumes
    union() {
    }
} // end of the difference
```

Programming

OpenSCAD contains a programming language with loops and mathematical functions.

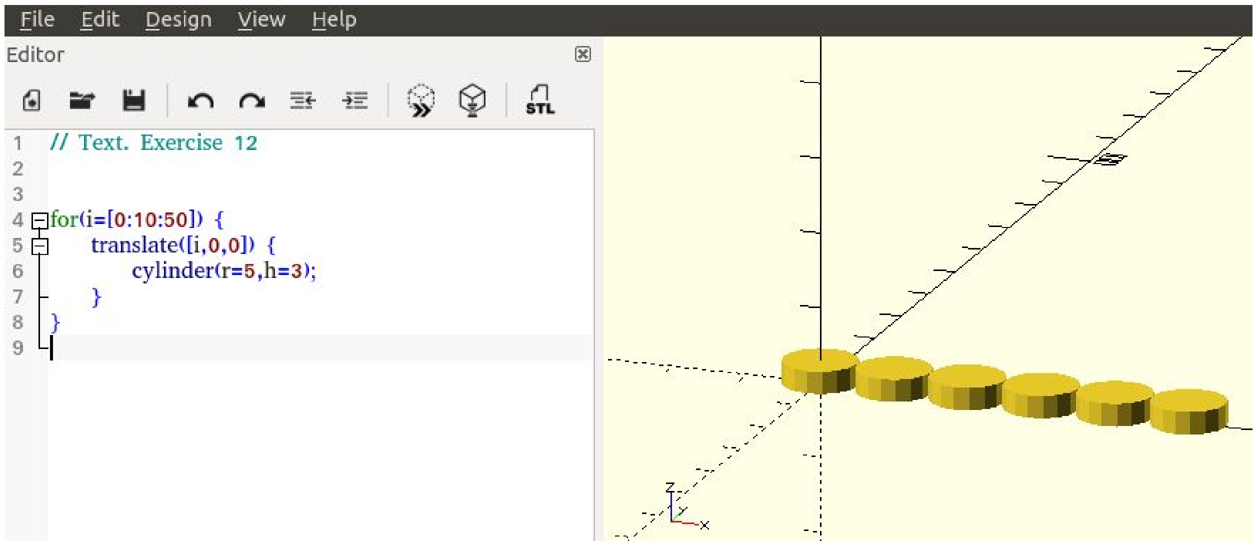
Here is the for loop for repeating shapes.

```
for(variable = [start : increment : end])
```

Exercise 12: Loops

Type into the editor window:

```
for(i=[0:10:50]) {
   translate([i,0,0]) {
      cylinder(r=5,h=3);
   }
}
```



Debugging

OpenSCAD contains several useful commands which help in debugging a model. Placing one of these characters at the start of a line will show or hide that part of the model.

Modifier Characters

*	disable
!	show only
#	highlight / debug
%	transparent / background

The echo() function will write any calculated value into the debug window.

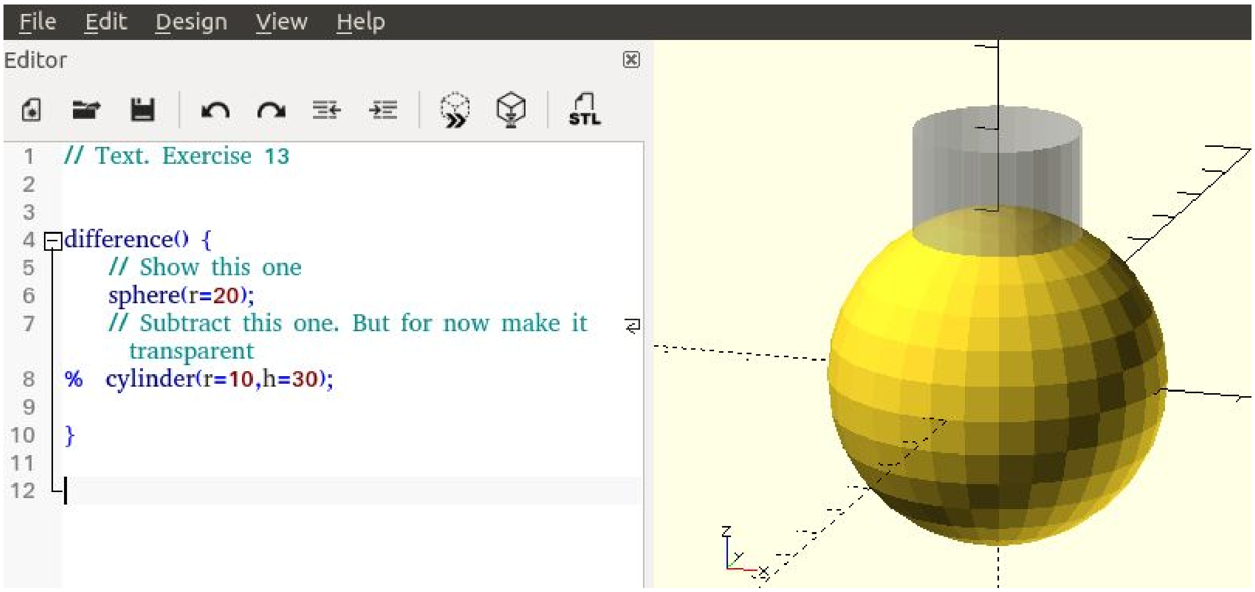
Exercise 13: Make part of the model transparent

Type into the editor

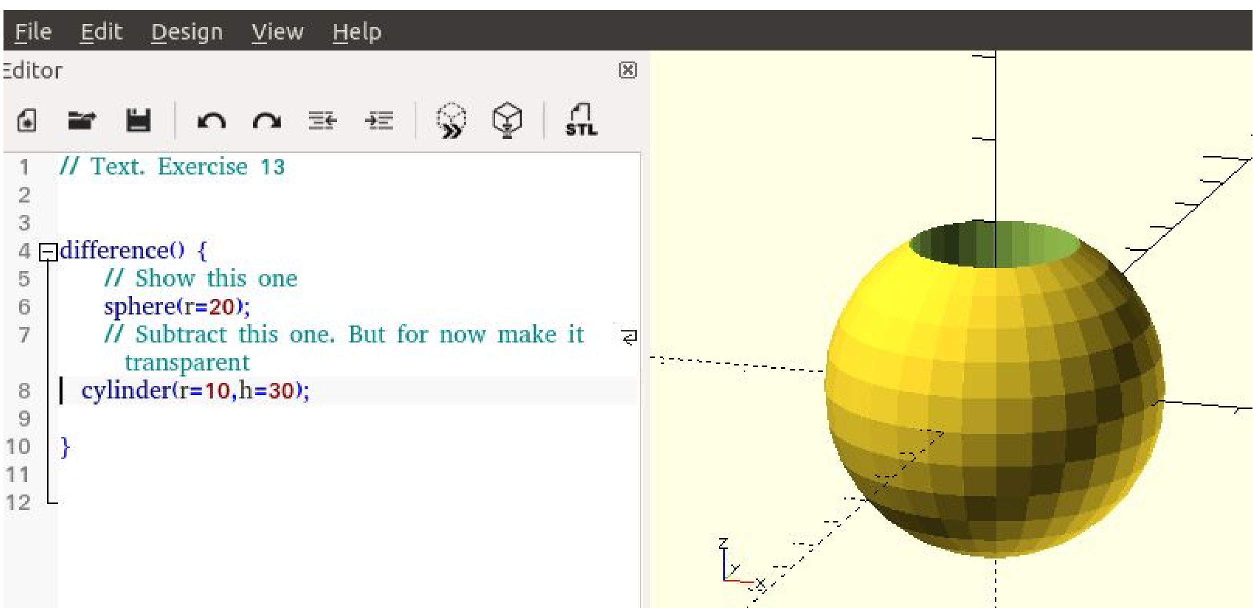
```
difference() {
   // Show this one
   sphere(r=20);
   // Subtract this one. But for now make it transparent
%  cylinder(r=10,h=30);

}
```

To produce:



Without the % modifier the result would be:



Advanced Features

Projecting a 3D shape onto 2D and exporting as an SVG file to use with laser cutters.

Hull and Minkowski transformations to make round corner boxes.

Mathematical functions (sin, cos, tan, atan, log…). cross() for vector cross product.

Custom functions. (module() {} or function() {})

Libraries. (use = “name of library of functions”)

Defining the number of sides in curved shapes $fn = k, or the angle of sides. $fa = k

Convert numbers to strings to display with the text() function using str().

Putting Multiple Shapes and Functions Together

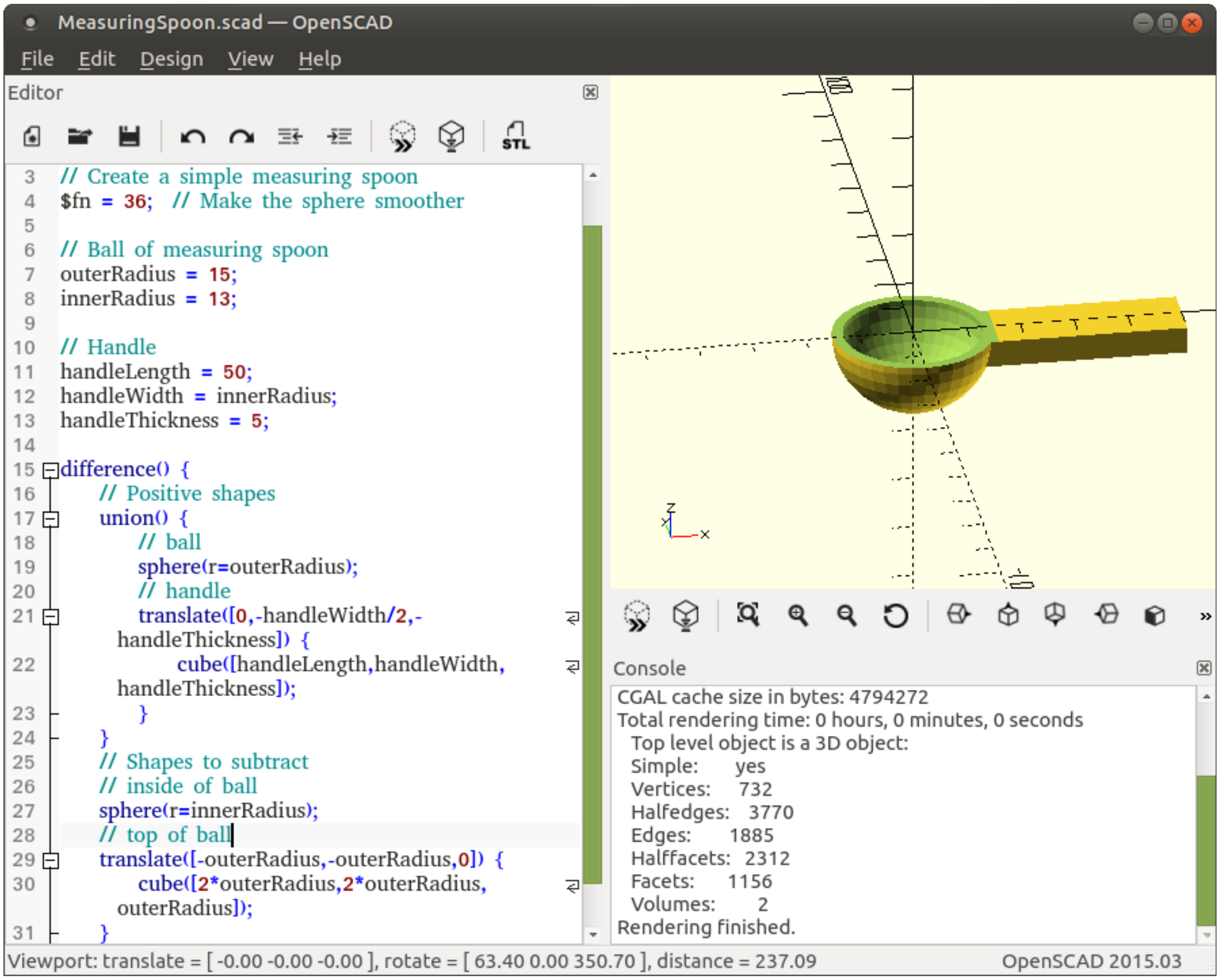
Exercise 14: Combining Features to make a teaspoon

Make a teaspoon of inner radius r=10, outer radius r=15, handle width = 15, length = 40, thickness of the handle =5.

Note: an inner sphere will be subtracted from an outer sphere. The top half of the sphere is subtracted off. The handle is a box (cube).

First, define variables for the inner, outer radius, handle width, length, thickness.

1. Define a difference which will enclose a union
    a. Create a union of the outer sphere and the handle
        i. Create a sphere at the origin of the size outer radius. (Default position is origin at the center).
        ii. Create a “cube” of the measurements of the handle. It will have to be translated to line up with the center of the sphere.
    b. Subtract off a sphere of the size of the inner radius.
    c. Create a box (translated) to subtract off the top of the sphere.


MeasuringSpoon.scad — OpenSCAD
File Edit Design View Help
Editor
// Create a simple measuring spoon
$fn = 36; // Make the sphere smoother
// Ball of measuring spoon
outerRadius = 15;
innerRadius = 13;
// Handle
handleLength = 50;
handleWidth = innerRadius;
handleThickness = 5;
difference() {
// Positive shapes
union() {
// ball
sphere(r=outerRadius);
// handle
translate([0,-handleWidth/2,-handleThickness]) {
cube([handleLength,handleWidth,handleThickness]);
}
}
// Shapes to subtract
// inside of ball
sphere(r=innerRadius);
// top of ball
translate([-outerRadius,-outerRadius,0]) {
cube([2*outerRadius,2*outerRadius,outerRadius]);
}
Console
CGAL cache size in bytes: 4794272
Total rendering time: 0 hours, 0 minutes, 0 seconds
Top level object is a 3D object:
Simple: yes
Vertices: 732
Halfedges: 3770
Edges: 1885
Halffacets: 2312
Facets: 1156
Volumes: 2
Rendering finished.
Viewport: translate = [-0.00 -0.00 -0.00], rotate = [63.40 0.00 350.70], distance = 237.09
OpenSCAD 2015.03